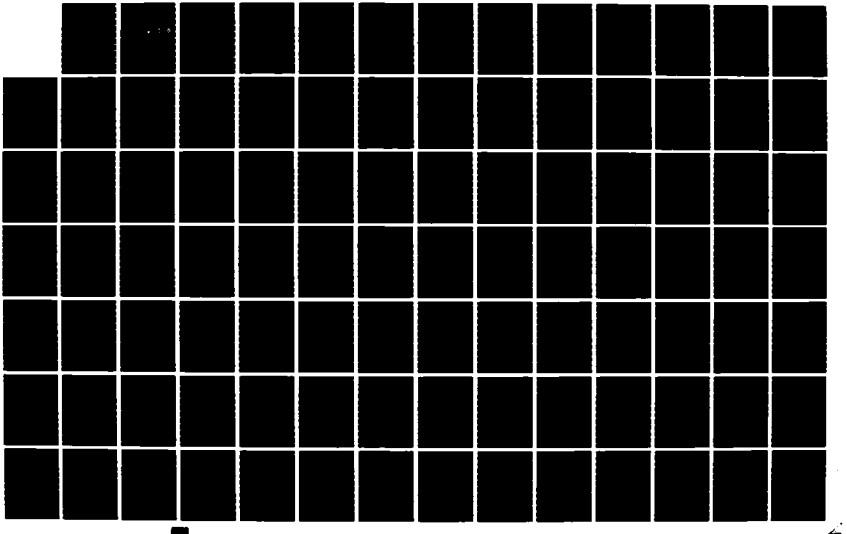
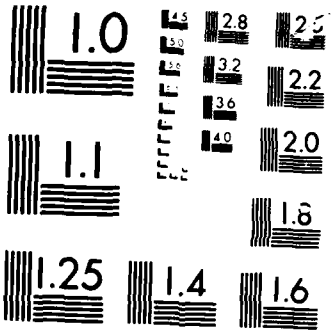
AD-A169 263 EFFECTIVE MICROCOMPUTER MANAGEMENT: AN EXECUTIVE LEVEL GUIDE(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M W SKAHAN 27 MAR 86 1/2

UNCLASSIFIED F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EFFECTIVE MICROCOMPUTER MANAGEMENT:
AN EXECUTIVE LEVEL GUIDE

by

Michael J. Skahan

March 1986

Thesis Advisor: Michael W. Spencer
Co-advisor: Jack W. La Patra

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE AD-A169 273

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): 54	7a. NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943	7b. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

EFFECTIVE MICROCOMPUTER MANAGEMENT: AN EXECUTIVE LEVEL GUIDE

12 PERSONAL AUTHOR(S)
Michael W. Skahan

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1986 March 27	136

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	ZENITH, MICROCOMPUTER, GENERAL SYSTEM THEORY, STRESS, MICROCOMPUTER MANAGEMENT, MICROCOMPUTER SURVEY

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A major hardware buy of Zenith microcomputers and peripherals was begun by the U.S. Navy in 1983 to equip squadrons, afloat units, and staffs with desktop computer capability. This contract provided considerable savings in hardware acquisition costs and the potential, with proper software, to radically improve the effectiveness of individual commands. This improvement has not been realized in many commands, due mainly to a lack of understanding of what a computer system is, how to ensure effective applications, and how the system should interact beyond the command. This study, aimed at the commanding officer, discusses management of a microcomputer system. It includes basic microcomputer capabilities and limitations, economics considerations, program requirements, training, and recommendations for enhancing microcomputer effectiveness.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL: Norman F. Lyons	22b TELEPHONE (Include Area Code): 408 646-2666	22c OFFICE SYMBOL: 54Lb

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted; All other editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

Effective Microcomputer Management: An Executive Level Guide

by

Michael W. Skahan
Commander, United States Navy
B.S., U. S. Naval Academy, 1970

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1986

Author: Michael W. Skahan

Approved by: Michael P. Spencer, Thesis Advisor

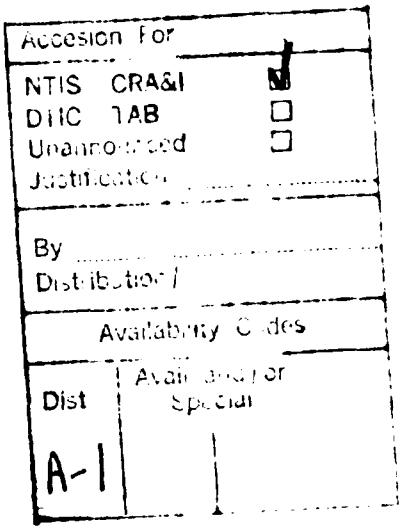
Jack W. LaPatra, Co-Advisor

W. R. Greer, Jr., Chairman,
Department of Administrative Sciences

Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

A major hardware buy of Zenith microcomputers and peripherals was begun by the U.S. Navy in 1983 to equip squadrons, afloat units, and staffs with desktop computer capability. This contract provided considerable savings in hardware aquisition costs and the potential, with proper software, to radically improve the effectiveness of individual commands. This improvement has not been realized in many commands, due mainly to a lack of understanding of what a computer system is, how to ensure effective applications, and how the system should interact beyond the command. This study, aimed at the commanding officer, discusses management of a microcomputer system. It includes basic microcomputer capabilities and limitations, economics considerations, program requirements, training, and recommendations for enhancing microcomputer effectiveness.



THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

In 1983, in a joint $30 million acquisition program with the U.S. Air Force, the U.S. Navy started purchasing the Zenith 100 series microcomputer in order to equip all commands with a desktop computer. Individual commands were given purchase ceilings to acquire the hardware (and software, if still below their limit), but little formal guidance or training was provided. This left each command to struggle on its own to best employ the equipment and software. In order to get the most out of the microsystem, the commanding officer must know what the capabilities and limits of the system are and how to manage that resource to improve his command readiness.

It has been said that "Knowledge is Power." The intent of this paper is to provide a basic level of knowledge in computer systems to mid and upper level management so that they may, through understanding, more fully employ the microcomputer in the management of their unit.

In solving a problem it is beneficial to take a systems approach in analyzing and thinking about that problem. The methodology of the "General System Theory" is discussed in Chapter II, and stresses the importance of establishing objectives or goals and defining the boundaries of the problem at hand as a first step in the solution prosess. This is then followed by a discussion of what makes up a business computer system. The micro system in this case is not just the computer keyboard and screen. Much of the dissatisfaction with the current microcomputers can be traced to such a limited view. The system will be shown to be far more complicated than that. Growth of the system and certain identifiable stages are then looked at.

Chapter III provides the upper level manager with an overall view of computers, including simplified operating theory, capabilities, ad limitations. The emphasis is on microcomputers and especially the Zenith 120 System, which is being supplied to Navy commands. In most cases these are the first microcomputers these commands have owned.

Software programs that can be applied to do a certain task are at the heart of the current microcomputer problem. It is important for the executive to understand what economic factors play in procurring such software. When software is to be produced in house, certain standards must be applied, again because of proven economic impact. Following the design considerations, the manager is provided with both ergonomic and psychological considerations. The implications of physical and data security conclude Chapter IV.

As part of the research for this paper, a survey of selected Zenith users was conducted. An analysis of this survey as well as information from NARDAC, Wing and Type commanders is provided.

Descriptions of current software in use , as well as a complete listing of a personnel management program are included as examples of possible programs which would be useful to other commands.

II. BACKGROUND

A. THE GENERAL SYSTEMS THEORY

The general systems theory provides a view point from which to observe a selected system. A system is described as being

> ...made up of sets of components that work together for the overall objectives of the whole [Ref. 1: p. 1'].

This is accomplished by feeding inputs to the system, having that system perform a transformation process, and outputting a new or changed product. The theory is part of a whole class of approaches to managing and planning that attempt to tie together, on a broad level, as many aspects of a problem as possible. [Ref. 2: pp. 7-8]

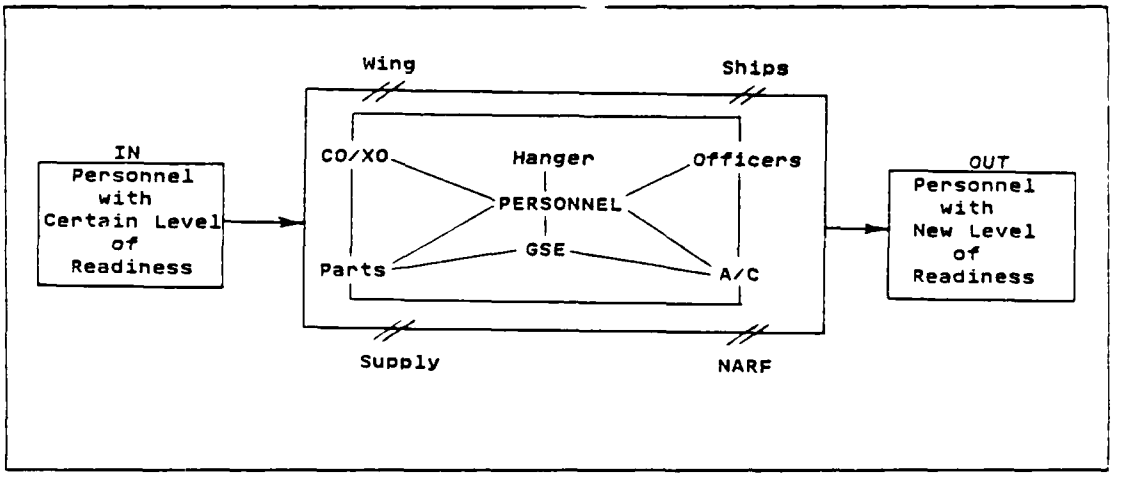


Figure 2.1 Model of a Squadron

Figure 2.1, adapted from Wetherbe [Ref. 3: pp. 10-12], shows a possible version of this input - transform - output concept. In this case it is interacting with the environment and so is termed an open system. Other attributes

would include being adaptive and organic, that is, it depends on the environment for new materials, energy, and information. The interaction may change over time and so can not be entirely predictable. In a squadron, people reporting aboard generally have a lower skill level than when leaving three years later. Demands of the environment (Wing, etc.) will change as that system itself interacts with its own environment. The emphasis and demands of one period will fade and be replaced as the larger system experiences changes. The availability of spare supply parts may change as the supply system is affected by its structure and environment. By being adaptive, the squadron can respond to changes and continue to operate. By approaching the problem from as many aspects as possible, improved results can be realized, as the subsystems involved will be equally stressed [Ref. 4: p.7].

A further refinement in the system model requires some sort of control. By allowing a method of feedback, the system can react to changes in a predefined manner and attempt to recover from movement in an undesired direction. It is not always necessary that the feedback be bounded on all sides, such as an engine governor which seeks to maintain one specific speed. Often a correction is only desired in one direction, like increasing office productivity. The manager, in setting such a control, must still keep the system as a whole in mind as other undesireable conditions may surface in a different area.

The short term gains may be offset for example, by long term losses due to worker dissatisfaction. On the other hand, the short term gain may be all that is important to the manager and he may demand only that. It is important, though, that the view of the system be from the highest level. As seen in Fig. 2.2, what may appear at one level to be obviously beneficial may, in fact, be detrimental to the

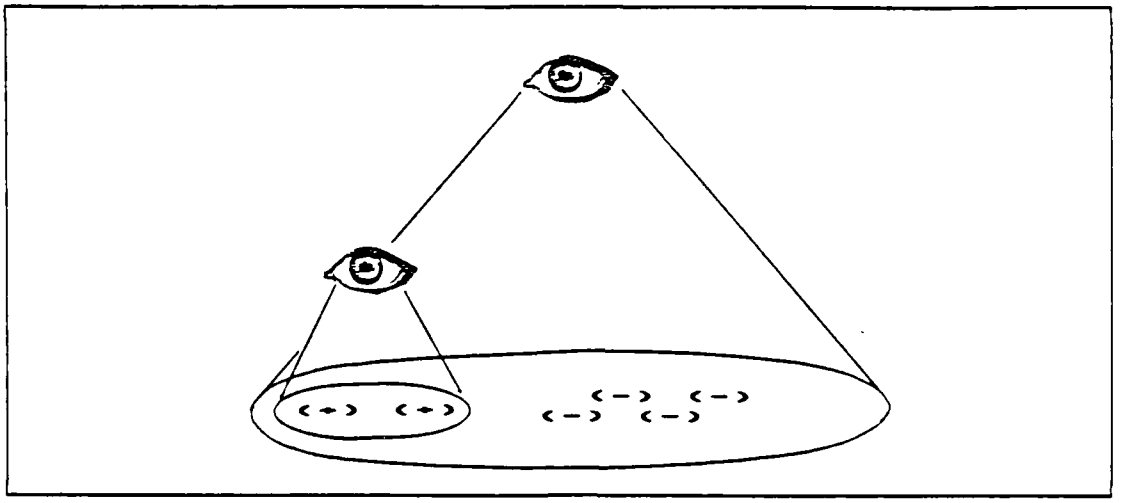


Figure 2.2 View of the System Makes a Difference

system as a whole. The good planner has to be imaginative and willing to take chances in playing out possible scenarios of what may happen in the future. Sometimes there will be conflicting objectives. He must be alert to avoid quick, easy solutions to solve a single problem that may later prove unsound for the organization as a whole. Each alternative must be weighed against achieving the desired overall goal, and how alternative courses of action may lead toward or away from stated objectives. A thorough understanding of the system boundaries is essential since the environment has a strong impact on the ability to reach the objectives. By knowing the environment, a better evaluation of alternatives can be made. As the leader is probably aware, a sound plan may still be very distasteful to some people in the organization. In this case, the goal of a subsystem may not have been fully considered, and when reviewed, may lead to a new alternative. [Ref. 1: pp. 169-173]

The system description can be refined for a squadron as:

1. The measure of performance, as defined by the manager (i.e. the C.O.).
2. The environment, as constrained by the level of technology, degree of training, deployment commitments, and other outside forces.

3. The resources in supply and personnel.
4. The components or subsystems as established in the squadron organizational structure.
5. The management, or decision policies of the C.O. and his degree of commitment in each area toward the objective. [Ref. 1: p. 65]

In designing the system, it is important to first establish the criteria or objectives of the system. The system itself is determined further by deciding what entities (smallest subsystems) to include and how to structure the entities [Ref. 5: pp.84-109]. In the case of a typical helo squadron, inclusion would be selecting certain people (entities) with known skills (attributes), to form a deployable aircraft detachment (objective), in the squadron (system). The structuring would be the mix of talent on the det. The inclusion process is, to a large degree, determined by the style of the manager and how he defines the objectives of the system.

The organization, or squadron in this case, is not formed spontaneously, its structure is based on events rather than static physical components, each exercising wills of their own. The departments of the squadron, each division, and even the individual members all contribute to the overall shape and make-up of the squadron structure. This autonomy of parts is a major variable in a human social system and causes a corresponding increase in the number of degrees of freedom for that system [Ref. 6: p. 25].

The interdependent elements, and the interaction between them is essential for understanding a systems approach viewpoint, while communication and information flow must be related to the system structure to fully understand the organization under study [Ref. 7: pp. 76-82].

B. COMPUTER SYSTEMS

A computer system, in a Administrative environment of a squadron, is made up of numerous components which interact to satisfy a particular need. It is important to note that

the computer itself is only one of the components. It and its related hardware must be joined with software programs and information data. In the recent Zenith buy, only hardware and limited programs were purchased, leaving applications programs and the data to the individual command to produce. But the system is still more than just hardware, programs, and data; it must also include the people who run it and the procedures they must follow. These five parts successfully describe a business computer system, and each must be understood for the successful integration needed to satisfy the desired goals of the system. [Ref. 8: p. 22]

The hardware is normally the most obvious part of the system. The terminal provides a method of input to the computer or specifically the central processing unit (cpu) and associated circuits. The screen can echo this input or can display the completed output visually. The disk drives provide a rapid input method and can also store the output magnetically. The output could also be sent to a printer for permanent, readable copy.

The programs provide a way of customizing the computer to complete a certain task by providing instructions for it to perform. In this way a general purpose machine, like the Zenith 120, can be made to do a number of tasks that is limited only by the number of programs available. The basic functions of the bare machine are essentially limited to adding, subtracting, storing, and comparing. A program merely causes the computer to perform these functions in a certain sequence. By carefully constructing and combining these simple functions, the software programmer can write instructions to build hundreds of different applications programs that will all run on the same piece of hardware.

The programs fall into two broad categories: systems programs and applications programs. Systems programs, generally supplied with the microcomputer hardware, provide

fundamental housekeeping for the computer such as placing the character "A" on the screen when the "A" key is pushed, properly writing information on a disk when a "save" is done, and showing the directory names of programs on a disk when "DIR" is typed. The other category, applications programs, is aimed at solving a particular problem, such as drawing a pie chart of OPTAR expenditures or giving a printout of the squadron's recall bill. Programs available commercially solve general business type problems, such as word processing. Problems unique to one command or one community would have to be custom written with those particular needs and problems in mind.

Data is the third component of a business computer system. This is the information that describes all the facts needed to solve the specific need. The data must be in a certain format to be compatible with the application program in use and it must be entered and stored so that that program can use it. The descriptions provided by the data must accurately reflect the real world situation or the resulting output will also be flawed. The types of data can be labeled the same way as hardware: input data, processing data, output data, and stored data. [Ref. 8: pp. 22-28]

People and procedures are the last two components. The procedures tell people how to enter data, what to do when errors occur and how to operate the computer. The key component, people, includes the users who actually operate the microcomputer, the programmers, and the clientele who use the end product. The manager must integrate these five components and consider them all as part of the system or benefits will not be realized.

C. STAGES OF COMPUTER GROWTH

When a computer system is introduced into an organization, its growth and, significantly, the way it should be managed, can be divided into identifiable stages. The

stages relate expenditures and growth to time and experience in system development. Figure 2.3 shows the characteristic S-shape, beginning with the introduction of the system, rising with expansion, slowing with formalization, and finally flattening out at maturity [Ref. 9: pp. 76-88].

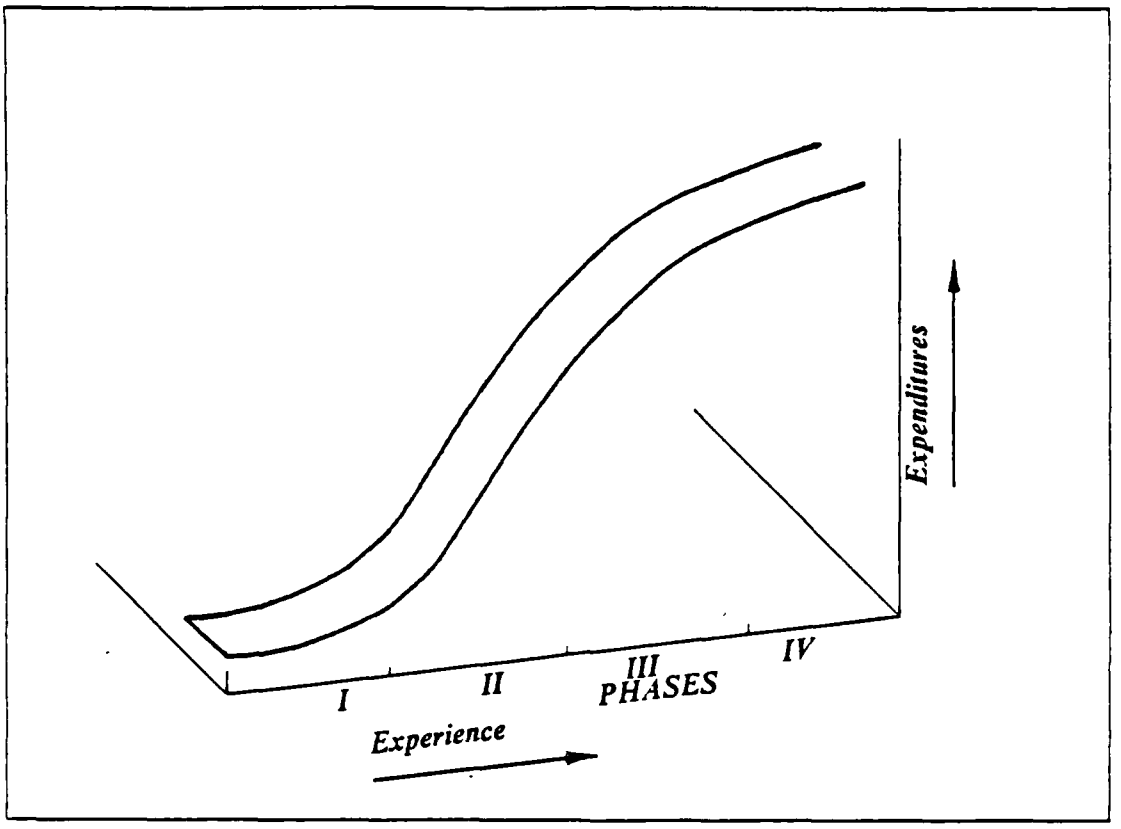


Figure 2.3 The Four Stages of Growth

The four stages shown in Figure 2.3 were later expanded to six [Ref. 10: pp. 115-126], but the relevance of the theory can be adequately explained using the original form.

Stage 1 is initiation. The long-term effects on personnel and the organization are rarely addressed by management at this point. Responsibility and location of the computer usually go to the department with the first or biggest application for it. This may present a problem

later when the system is more broadly used. The first department becomes protective of the system, and the department head may see it as a source of personal influence and power; giving it up may not be acceptable to him.

The first stage may also cause some anxieties to appear as the new technology is introduced, Chapter IV treats this problem more fully and provides the manager with symptoms and remedies to minimize the disruption that has often been observed in similar settings.

The first stage is characterized by simple, cost or labor reducing functions. Management of the system is lax, not many controls are in place, and priorities are generally first-in/first-out.

Stage 2 is expansion. Broader and more advanced applications are used as more departments try to exploit the "power" of the computer. In a mainframe environment, the system will be able to absorb many new users, but with a micro, one user ties up the whole system--input, processing, and output. More users mean less time for each. This period is marked by unplanned and mostly unsupervised growth. Projects multiply, but with little coordination. Duplication of effort and incompatible data grows. The most knowledgeable personnel take on projects they see as well intended, although in reality may be only personally interesting and rewarding, but not in the organization's best interests. It is important for the C.O. to anticipate this stage and while encouraging new uses, experimentation, and growth, direct these activities in a productive and farsighted manner. His department heads should be aware of what to expect at this stage and work together to ensure coordinated projects that satisfy individual, as well as squadron needs.

Stage 3 is formalization. This is the time that stricter control methods are put into force. It may finally

be realized that a senior chief or a lieutenant-commander can't be used as a resident programmer and primary terminal operator. The cost is too great. Formal control steps must replace the informal relationships. What normally occurs in stage three, should ideally be addressed early in stage two, but too often the control stage is not anticipated and becomes reactionary. Selection of projects and standards for the projects should be addressed early. All departments should be involved and the C.O. should be kept aware of the progress in each major project.

Stage 4 is maturity. This is the point at which the system will be able to return continuing economic benefits. Both users and senior management understand the technology and there is a comfortable feeling with the applications. This can actually be a difficult time for the senior manager, who must keep this harmony, but at the same time, keep up with newer methods. If no changes are made, his organization will fall behind technologically. If new changes are introduced, he disrupts established harmony. What is needed is a plan to flatten out the curve and prevent sliding too far back on the curve. [Ref. 9: pp. 87-88]

Although Nolan's model is a widely accepted frame for DP growth, Goldstein and McCririck [Ref. 11: pp. 309-324] offer an alternative for the S-curve. Their conjecture is that the initial flat growth (initiation), followed by rapid expansion (Nolan's Stage 2), and then a flattening period (control), was due to the introduction of new technology and national economic prosperity during those middle years. Nolan's maturity stage was not questioned. Despite this, the model has remained a standard for industry measurements of DP growth. The exact fit of the Nolan model with respect to microcomputers remains to be proven, especially given the explosive advances in the last three years. The basic

growth pattern, however, appears logically sound and at present remains a good model for comparison when the manager looks at the state of his own computer growth.

III. COMMAND EXPECTATIONS FOR THE MICROCOMPUTER

A. BASIC COMPUTER INFORMATION

Depending on career paths and educational backgrounds, many top level managers have reached their positions without exposure to the fundamentals of how computer hardware and software work together. Knowledge of this information allows the manager to better understand the capabilities and limitations of the microcomputer system and, therefore how to to better employ those assets within the command.

1. Binary Counting

The computations done by the computer, the numbers and letters stored in it and all that is stored on the floppy disks is done with binary digits called bits. Like a bulb that is either on or off, only two states exist in this numbering system. In this base 2 system, each digit can provide only two pieces of information. By combining bits, more complex pieces of information can be represented. In an 8-bit machine, eight bits are combined to form a byte. Each byte is a combination of eight 1's or 0's, so the total number of different patterns is 2 raised to the 8th power, or 256. This allows all the letters, upper and lower, all punctuation, and all 10 digits to be represented and still allows for extra characters, usually called control characters. If the word size of the machine is increased to 16 bits then it can represent 65,536 different patterns, making it much more versatile. The Zenith comes with two CPUs, one for 8-bit and one for 16-bit modes. The bits in the computer are represented by either direction of current flow or by voltage level (usually 0v = 0, 5v = 1). On the floppy disks, the brown oxide coating is very similar to audio cassette tape coatings. The bits here are represented by the direction of magnetization at selected points on the disk. [Ref. 8: pp. 31-32]

2. Memory

The Zenith computer is supplied with "random access memory" (RAM) boards that provide 256 Kbytes of internal storage space that may be written to by the user or by the program. One K, or kilo, is (2**10) or 1024, but is usually rounded to 1000. By adding more boards inside the computer, memory can be increased to 768K. The problem with RAM is that when the computer is turned off, what was stored is lost. The disk storage system allows the RAM to be copied onto a floppy disk and saved. The computer also has "read only memory" (ROM) which holds permanent instructions to tell the computer how to do simple start-up and other general operations, such as reading a more complex program off a disk. The instructions on the integrated circuits holding ROM are coded by the manufacturer and cannot be changed. The CPU is the part of the computer that actually does the work. The Zenith has an 8-bit 8085 and a 16-bit 8088. These CPUs are each made up of a timer (control unit), an arithmetic and logic unit (ALU) to execute instructions, and a memory section to hold instructions and data. Depending on the program only one CPU will be active at a time.

3. Disk Drives

The Zenith is normally supplied with two floppy disk drives, refered to as A and B. This permits a program in A to access data in B. This arrangement makes more room available for data and keeps the data isolated from the program disk. The drives each contain a motor to spin the inner part of the disk and a stepping motor to precisely position a magnetic head on the disk's surface to read or write information. A new disk, when it is formatted, is magnetically divided into concentric rings called tracks and each track is divided, like a pie, into sectors. Each sector typically holds 128 or 256 bytes of data. The data

is arranged on these sectors in a known pattern permitting later retrieval of the desired information. A full disk will hold about 320 Kbytes of information, which is equivalent to over 100 pages of text. The floppy disks must be rated at double sided and double density. Although less expensive single density disks and even single sided disks can sometimes be made to work, disks which have been certified by the manufacturer at the higher capacities will provide more reliable service.

The disk spins at about 300 RPM and the stepping motor can quickly move in and out to reach any part of the disk's surface so the operation seems almost instantaneous. However, the computer's CPU can work much faster than the mechanical disk drive so it still must wait while information is read or written. A program, when it accesses the disk, will noticeably slow down while it writes or retrieves information. The actual delay is due to three factors: the head must move to the proper track, the disk must rotate to the proper sector, and then the data must pass under the head. This time, measured in 10's of milliseconds, can become significant in terms of user wait time when multiple, consecutive reads are requested, such as in data base queries. Similar delays would be expected when writing information to the disk. A more efficient but higher priced system uses a hard disk or "Winchester" drive. This is a sealed, multi- platter drive that spins faster and can pack the recorded data tighter on the disk. This results in faster access times and storage equal to ten megabytes or about thirty floppy disks. [Ref. 12: p.21]

4. Printers

The two types of printers supplied with the Zenith are dot-matrix and daisy wheel. The daisy wheel has preformed letters on a removeable element and produces crisp, high-quality output, but at a relatively slow rate.

The print elements can be interchanged so that different type styles may be used. The dot-matrix printer uses a head with a vertical row of pins that are controlled by the computer to produce characters as the head moves across the paper. The output is fast at 100-200 letters per second, but each character is formed of a 5x7 dot matrix pattern which gives the familiar "computer output" quality and so is not suited to professional correspondence. Newer printers are now available that make two passes over a line to more completely fill in the characters and come close to "letter-quality" output. The big advantage of dot-matrix printers, besides fast output, is their ability to produce graphics and to change font styles on-the-fly. Charts, graphics, and pictures can be printed as part of a text document and the text itself can be of different styles, even on the same line of print. A third type of printer which uses a laser scanned drum, can produce output as good as a very high quality photocopy. Printing a whole page at a time, it has the advantage of business-quality output and graphics, but cost is three to four times that of a good dot-matrix printer.

5. Modems

Modulators/demodulators provide the necessary connection from a computer, via telephone lines, back to another computer. Software programs such as BestTerm handle the interfacing, dialing, connection, and upload/download requirements for transferring data and programs. Modems are available at several baud (switching) rates, but 300 and 1200 baud are most common. At 300 baud the transmission rate is slow enough to be read by the user as it comes in, however the 1200 baud speed is four times faster and proportionately reduces the required connection time. The ability to remotely download information opens up a broad wealth of opportunities. The Naval Data Analysis Center in Washington

presently has numerous programs available for download to fleet commands. Within a single command, multiple micros could be networked together and allow data to be shared among the different departments. In another area CNAVRES now sends out reporting orders for reserve personnel via modem to a printer at the parent command, thus reducing to a few minutes what had previously taken weeks.

B. APPLICATION PROGRAMS

The software programs provide the computer with the flexibility to be used for many different types of jobs. This section describes the programs that are being supplied to Navy units as part of the standard start-up package and details their major functions, strengths, and limitations.

1. Disk Operating System (DOS)

The DOS is the program that is loaded into the computer first. It contains the instructions for the disk drive functions such as formatting or copying. With its two processors, the Zenith can run both 8-bit CP/M programs and 16-bit MS-DOS or Z-DOS programs. CP/M stands for "Control Program for Microcomputers" and has, over the last ten years, become the standard operating system for 8-bit microcomputers. In addition to the hundreds of commercial software programs available, Froehlich [Ref. 13: pp. 65-385] has cataloged over 5000 programs that are available in the public domain. For the most part, the authors who wrote these programs did so for an immediate need at the time but some may be adaptable to current squadron needs.

Z-DOS was the original DOS supplied from Zenith, but the more universally used MS-DOS is now the favored and primary operating system, giving a rich variety of commands and allowing the machine to run many commercially available text format programs. The major functions of DOS are loading, saving, and copying files. The MS-DOS manual, in its 600 pages, provides extensive information on getting

started with the system, the primary features and commands, and even on program development. Three special commands are worth noting:

1) The RDCPM command can read CP/M formatted disks and copy them into MS-DOS formatted disks, making more programs available and standardizing the format.

2) The MS-DOS command CIPHER can encrypt files or create encrypted copies of regular files, making them unreadable to others not knowing the original key word. Without knowledge of the key word, files cannot be run, listed to the screen, or printed [Ref. 14: pp. 11.64]. Although this would not be acceptable to hold unlocked classified material under OPNAV 5510 series, other proprietary information like fitness reports or enlisted evaluations could be stored under this protection scheme.

3) The RECOVER command can salvage the useable portions off a disk which devélopes bad sectors. Bad sectors are areas with magnetic or physical damage to the disk that makes that part of the media impossible to read. This feature may be able to salvage most of a damaged disk and not require recreating the files manually. [Ref. 14: p. 11.264]

2. Disk Library Organization

As the number of files created by a squadron increase, it is important to maintain some systematic ordering scheme for all the disks and files. DOS will allow many filetypes--program files, command files, text files, and data--to reside on the same disk, but it is far more productive and less confusing to users if an organized system is established. The command's disks will normally fall into the following categories: system disks, working disks, archive disks, backup disks, and library disks. [Ref. 13: p. 53]

The system disks hold the operating system, command files, and other frequently used files. This disk is mounted in drive A and should contain the word processing, data base, spreadsheet, and language programs.

Working disks are those created while using the systems programs such as text files from WordStar or data for dBase II. These disks would be used in drive B.

Archive disks are used to save old files for historical purposes or for infrequently used files.

Backup disks are copies of system disks, working disks, and archive disks. They provide a means of recreating data that might be destroyed if an original disk cannot be read for some reason. The backup should be stored at a location that is physically separated from the original so that both copies are not lost in the event of theft, fire, or other disaster.

A backup copy of all command disks is cheap insurance, but creating them at too frequent an interval can be costly in terms of manpower. The rule is to leave only as much as you are willing to lose, unbacked-up.

Library disks are a cataloged collection of all software held by the command.

DOS allows a file to have a filename of up to eight letters (and/or numbers) followed by a period and then an extention of up to three more characters. The labeling system can make use of this somewhat limited naming ability by describing the disk with its subject plus a number and using the extender to identify the type of file. Thus possible extenders could be: SYS for system, DAT for data, BAS for basic, COM for command, EXE for executable, ASM for assembly, and BAK for backup copy. For instance, AIRCRFT8.DAT might be a file holding flight hour data on aircraft 08. Each disk itself should have a label marked with a felt tip pen (to prevent disk damage) to describe and number the general contents, and a file such as DISKS001.DOC should be maintained to keep track of all the files on each disk in the command's collection. Since the number of disks used by an organization multiplies very rapidly, it is advisable to set up a filing system as part of the introduction phase. [Ref. 13: p. 54]

3. Data Base Management Systems

A data base management system is a program that allows the user to organize and quickly access large collections of related information. After word processing, this type program can probably provide the most benefits for a command. Appendix B, for example, is a driver program written for dBase II that the squadron Admin officer might use. Producing printed lists pertaining to members of a squadron, it has a menu with choices that include security listings by clearance, a recall bill, a seniority listing, and a PRD listing. The basic information about each squadron member is held in a data file which the program selectively accesses to pull out only the desired information. Thus the seniority listing would not include home phone number, but the recall bill would. The maintenance department might use a similar program to track aircraft and engine hours, time left before periodic inspections, and even print out an up-to-date aircraft custody report. The training officer might keep all audio/visual and training devices cataloged and could print an inventory or "PMS due" list. In operations, all pilot qualifications and expiration dates could be maintained and quickly printed. The more complex the collection to be controlled, the more a computer can help. [Ref. 15: p. S-2]

Although it would be possible to write a program in BASIC, COBOL, or Pascal that could search through a data file to find a piece of information, the program would need to be rewritten for every new bit of information desired. The major advantage of a database management system is that these search functions are built in so each search is initiated with a simple English-like query such as, "list all for name = SMITH." All the Smiths in the command will quickly be displayed. Since dBase II is a relational data base, two data files can be used at the same time and the ad hoc query

could be narrowed to "list all for name = SMITH and dept = MAINT." The point is that special searches can be made without having to write and test a new program each time new information is desired. For known periodic requirements, like a recall bill, a special driver program similar to Appendix B can be created to allow even an inexperienced user to print out the reports. Every step is menu driven so very little training is required.

The storage capacity of a floppy disk limits the size of an application file. Figure 3.1 shows the parts of a record.

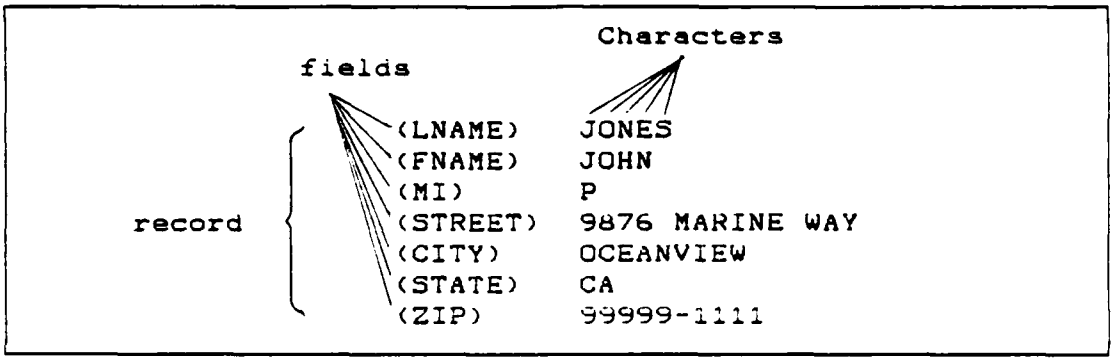


Figure 3.1 Parts of a Record

The records for many people would be combined into a file to be stored on the disk. The record itself is made up of different field categories and filled with characters. This example has seven fields in the record. To make room for longer names and addresses, 20 characters might be reserved in each field. This record would then require 140 bytes (one for each character) to store on a disk. Before searching for a last name, the records must be indexed (sorted) by last name. The index file would need about 22 bytes per record, so the disk would be able to hold about 2000 records with the name index. As more indices are used, less space would be available for records, even though dBase

II allows 65,000 records to be addressed. dBase II can handle 32 fields per record and 254 characters per field, but with a maximum of 1000 characters in any one record. Thus a data base describing a squadron by names, rates, addresses, phone numbers, dependents, and security clearances, will fit nicely onto one disk.

4. Word Processors

The Navy has relied on word processors for a number of years because of the flexibility in creating, correcting, revising, and expanding documents with the ability to then print out as many perfect copies as desired. Table I [Ref. 16: p.22] summarizes the advantages and disadvantages of a word processor.

TABLE I

ADVANTAGES AND DISADVANTAGES OF A WORD PROCESSOR

Advantages	Disadvantages
Productivity increases due to less retyping.	Cost is greater than a typewriter.
Work done in less time.	More training required.
Accuracy is increased (less repetition, spelling checkers, etc.)	New filing methods are needed.
More done with fewer people.	Computer must be in an up status to access files.
Less storage room needed.	More space needed for installation.

WordStar by MicroPro is a highly flexible word processor with a rich variety of commands that provide almost every function desireable in a word processor. It has selective settings of "help" levels, edit-while-printing, ability to read in files to text and automatic production of back-up files. It can also be used in conjunction with SpellStar (a 20,000 word spelling/vocabulary checker that can be augmented with user-specific words or acronyms) and MailMerge (a time-saver for repetitive documents). On the negative side, the commands are not intuitive to the average user. They may become second-nature to the frequent user, but must be memorized. The

majority of the commands require at least two or three key strokes (i.e. the control key and another key simultaneously, followed by a third key). The view of what the text will look like when printed is deceiving, especially when using the Special Effects commands. WordStar also does not show the size of a file, date of creation/revision or amount of space left on the disk or in memory. While editing, WordStar does not automatically realign paragraphs. The program comes with WordStar User's Guide but the fact that at least six other books like WordStar Made Easy [Ref. 17] have been published, underscores the sometimes tedious and confusing commands of WordStar. No one manual seems to encompass and easily instruct the novice user who is often told of two or three different ways to do the same editing function. The manager should keep these limitations in mind when directing training for new users.

The manager should also control the number of times a document goes back to the yeoman for correction and reprinting. There is a strong tendency for each level in the "chop chain" to submit a letter-perfect draft for review because it is "so easy" to put it back through the computer for one or two clean-up changes before the next superior in the chain gets it. A letter may be reprinted a dozen times before it reaches the C.O.'s desk for signature. The changes may be effortless, but time and manpower are clearly not being optimized.

5. Spread Sheets

The ability to set up an electronic spread sheet greatly simplifies and speeds up financial and other tabular calculations. The spread sheet is a grid of "cells" defined by rows and columns. Each cell can hold labels, numbers, or formulas, so that a full financial statement can be set up. The power of the program is the ability to fill cells with formulas, rather than hard values, which take values from

other cells to compute the defined value. Thus a change in price, for example, at one cell will be automatically computed in every other cell that uses that price. The manager can get extremely rapid answers to his "what if" questions. Examples:

(1) "If we double our flying hours for the next two weeks, how will it affect our available fuel money?"

(2) "Show me all those who will have the required time in rate before the next advancement cycle."

Whole worksheets or just extracted parts can be printed or saved to disk.

The supplied Lotus 123 is a popular and powerful spread sheet program that is relatively easy to set up and use. It has a grid size of 256 columns by 2048 rows. The "active area" seen on the screen is eight columns by 20 rows, but any point on the grid can be quickly reached, Lotus supports "macro" instructions which are stored keystroke sequences that make setting up a sheet, or template, much easier. This also allows a less knowledgeable user to learn to use the program more quickly. By pressing the ALT key plus a letter key, a string of preset commands is automatically executed. [Ref. 18: p. 112]

Lotus 123 also provides a limited data base management system and a graphing capability. The records of the data base are stored in matrix form with each record on a row and the fields of each record under a different column heading. With this capability the data does not need to be restricted to just financial data. Any information that needs to be tracked, sorted and retrieved can be entered.

The graphing feature of Lotus is particularly useful for the manager in spotting trends or anomalies in the matrixed data. Six sets of data can be graphed at once from the stored data. The pictorial record also makes the effects of "what if" questions easier to evaluate. Lotus can provide bar, stacked bar, pie, and line graphs [Ref. 18:

pp. 229-234]. The three parts of Lotus work together in what are called "interactive programs." Information entered once can later be used by any of the three programs.

6. Languages

There are five general types of languages used in a computer. Machine language is the lowest level and is written with series of binary numbers such as 11001001 for 8-bit computers. These numbers, which represent instructions and data for the computer, can be entered directly into memory locations in the computer. It is also possible to enter them in hexadecimal notation so that each 4-bit segment is represented by a number from 0 to 9 or a letter from A to F. The binary 11001001 can thus be shortened to C9, which is easier for the programmer.

The next level up is assembly language. This uses a mnemonic label for each instruction. LDA 3 would mean to immediately load register A (a storage location) with the value 3. Each line of the program contains similar abbreviated instructions. An experienced programmer can code a program with these mnemonics much easier and faster than with binary or hex numbers. When the program is finished, a special "assembler" program converts the code into a machine code "object program" for the computer to execute. [Ref. 19: pp. 52-55]

The three highest level languages rely on compilers and interpreters to produce code that the computer can use. A compiler takes a whole program written with English-like commands and compiles (transforms) it into a machine language program that can be run by the computer. A pseudo-compiler does the same, but produces an intermediate code first. The Z-BASIC supplied by Zenith uses an interpreter. The program is written by the programmer using the dialect, grammer, and syntax of BASIC. When it is run, the interpreter program takes each line of BASIC and translates it

into machine code. When a branching instruction is encountered, the interpreter searches from the beginning of the BASIC program until the named line number is found. These two factors cause programs written in BASIC to run very slowly, but the writing of the program is easier than in other languages. Z-BASIC also has built-in error checking codes to help the programmer in debugging a program during development. Additionally, it is supplied with a compiler program that allows programs to be developed and tested in BASIC and then be converted into object code that executes much faster. A side benefit to this is that the object code produced is itself unreadable and therefore secure, but it will still execute the program. The compiler, unfortunately, does not support all BASIC commands. [Ref. 20: p. 40-53] BASIC is probably the most popular language used on microcomputers, but its ease of learning and use also lead to important shortcomings of the language. BASIC code is written with a line number followed by a command or statement. The line numbers usually start at 10 and skip by 10's so that extra lines can be later inserted during the debugging phase without having to renumber the whole program.

A typical BASIC program looks like Figure 3.2 . This is a sample program that determines interest on a given principal. By placing the calculating portion of the code near the beginning, the interpreter can run faster because there is less line searching. The GOSUB and GOTO commands allow movement to other parts of the program, but this jumping around makes the logical process of the program more difficult to understand. The smooth left margin makes it difficult to pick out specific parts of the program. A longer program, covering many pages, would be very difficult to understand and therefore difficult to maintain even for the original programmer.

```
10 REM INTEREST BY YEAR
20 REM SET UP PROGRAM
30 GOSUB 200
40 REM PRINT PRINCIPAL AND INTEREST
50 PRINT
60 LET A=P*(1+R/100)^N
70 PRINT "YEAR = ";N
80 PRINT "AMOUNT = ";A
85 REM INCREMENT YEAR COUNT
90 LET N=N+1
95 REM COMPARE COMPUTED TO DESIRED YEARS
100 IF N>Y THEN END
110 GOTO 50
200 PRINT "IF YOU TYPE THE AMOUNT OF PRINCIPAL"
210 PRINT "AND THE INTEREST RATE PER YEAR, I WILL"
220 PRINT "SHOW HOW YOUR MONEY GROWS, YEAR BY YEAR"
225 REM GET INPUTS
230 GOSUB 300
240 RETURN
300 PRINT
310 PRINT "PRINCIPAL AMOUNT:";
320 INPUT P
330 PRINT "PERCENT INTEREST RATE:";
340 INPUT R
350 PRINT "NUMBER OF YEARS:";
360 INPUT Y
365 REM INITIALIZE YEAR COUNTER
370 LET N=1
380 RETURN
```

Figure 3.2 BASIC Program to Compute Interest

Contrasting the BASIC program with the Pascal program in Figure 3.3, it can be seen that the latter is written in a structured form. It is made up of four modules or "procedures" that each perform a specific task. Each procedure is identified by a name and also defines any values that will be passed to or from it when the program executes. Each procedure can use both global variables and with variables defined only within itself. This allows procedures to be written that are relatively independent rather than tied specifically to a single program. The lines are unnumbered and indented to form small pockets of logical operations. The actual writing of the program can be done on a word processor which allows easy editing, movement of blocks of code, and the insertion of previously developed procedures for reuse. This approach makes the design, development, and maintenance of the program easier than attempting to work on a single, monolithic program. In Figure 3.3 the main program is actually only four lines long, essentially calling on previously defined procedures.

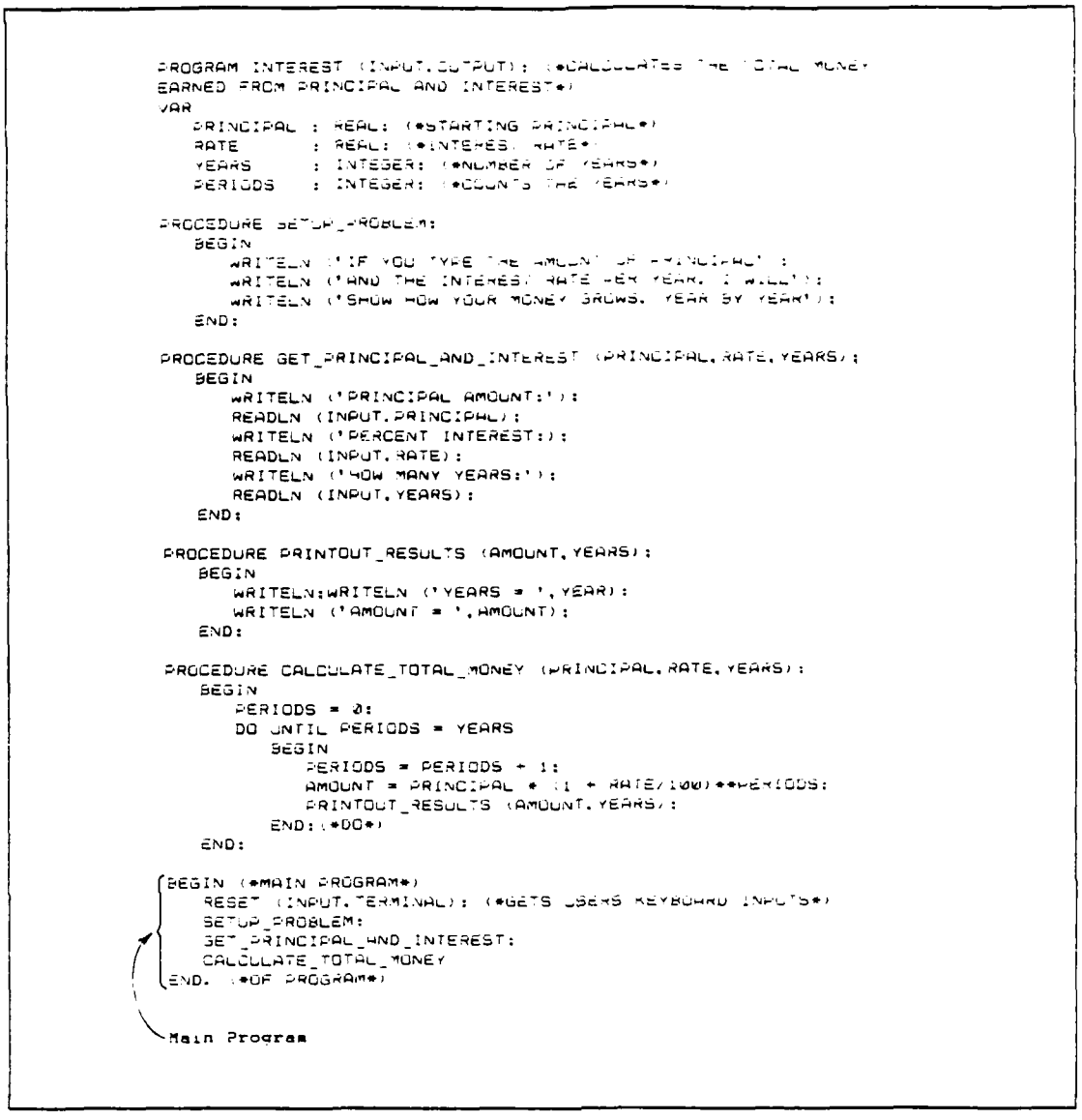


Figure 3.3 Pascal Program to Compute Interest

BASIC as well as Pascal and other structured languages like FORTRAN and COBOL rely on three syntactic forms to implement required logic for program formulation. Sequential, decision, and iteration provide the required structures as follows [Ref. 21: pp. 172-177]:

Sequential. The program executes each step in the order encountered.

Decision. Based on stated conditions, the program execution either continues or branches. This may be

an if/then decision or may be more complex based on a set of cases.

Iteration. The program loops through the same logic series a number of times while a count of the iterations is maintained until a selected value is reached.

IV. COMMAND POLICY AND MANAGEMENT OF MICRO ASSETS

A. ECONOMICS CONSIDERATIONS

The presence of a microcomputer in the squadron will not make or break the command, but having one that works well will give that squadron a great advantage in savings of manpower and ability of middle and top level leaders to better understand and therefore manage their areas of responsibility. There are costs and trade offs that must be considered when the decision to use a computer is made. It is a waste of time to establish a data base to track something in the squadron, unless there is also a commitment to spend the time and effort to keep it accurate and continually updated. There must also be a commitment to training for users, for managers, and for those developing special programs for the squadron. The actual development of computer programs represents a very great expenditure in terms of manpower. Programs which are developed haphazardly or with poor documentation can result in increasing rather than decreasing workloads. Methods of software analysis and design will be discussed in the next section, but assuming a command has resident members who have the skill needed to produce specified software, the commanding officer must still decide whether the costs of applying this talent can be justified by the results realized in the finished software.

Early and accurate prediction of software costs is difficult because of the nature of the product. Coded instructions are not a uniform commodity and are not even the end product, but only a means of accessing some other form of information. Software production requires cooperation between those who need the service and the person or group actually producing the program. And finally,

programming is a creative process that is dependent on the experience, talent, and dedication of its author.

There is a price to software whether it is produced in-house, purchased from a vendor, or contracted from NAVDAC. A common pitfall which leads to underestimating the cost of software is the assumption that coding is the only cost. More accurately, software should be considered to have a life-cycle with definite phases. Boehm [Ref. 22: pp. 36-38] describes this in eleven steps:

1) Feasibility. Defining the need, alternatives, costs, time requirements, and chance of success.
2) Requirements. Complete specifications and required functions.
3) Product Design. Specifications for software, control, data structure, and draft user's manual.
4) Detailed Design. Refinement of design to include all structures, relations, sizing, and algorithms for main modules.
5) Coding. Complete set of program components.
6) Integration. Properly functioning software products.
7) Implementation. Functional system including data conversion, installation, and training.
8) Maintenance. Fully functioning update of program.
9) Phaseout. Transition to successor program.
10) Verification and Validation. Was the program built right? Was it the right program to build?
11) Configuration Management. Milestone achievement and verification with each iteration.

Figure 4.1 [Ref. 21: p. 561] is a simplified view of development requirements in terms of time. The coding portion is only one third or less of the total effort. It also does not provide for maintenance of the program which has historically required up to fifty percent of the total software development effort [Ref. 22: p. 18].

A "Constructive Cost Model" (COCOMO) has been developed which estimates software development effort and cost solely as a function of the size of the software product measured

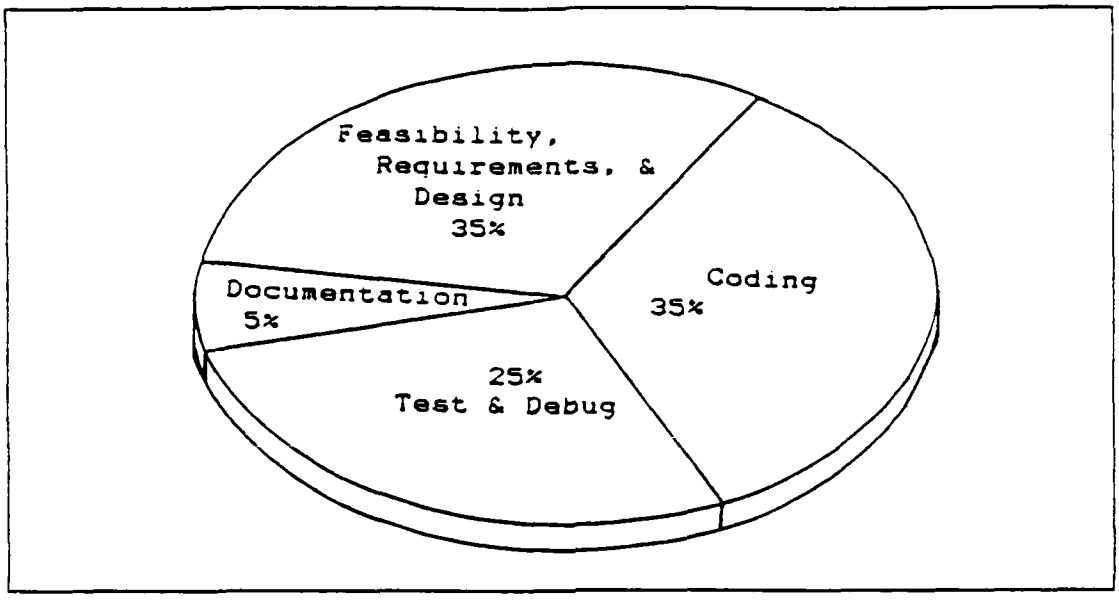


Figure 4.1 Typical Development Time Requirements

in number of coding instructions. The basic formula for determining time of development in man-months is:

$$MM = 2.4\ (KDSI)^{1.05} \qquad \text{(eqn 4.1)}$$

KDSI is the size of the program in thousands of delivered software instructions. One man-month is 152 hours of working time. If more than one person is assigned to the project, the time of development is not reduced linearly but can be computed by:

$$TDEL = 2.5\ (MM)^{0.38} \qquad \text{(eqn 4.2)}$$

By dividing the total man-months by the time of development, the recommended number of personnel can be established. [Ref. 22: pp. 29-70] While the basis of development time is the number of lines of code, this must be tempered with other attributes which include the product, the computer, personnel, and product management. The software products

will vary by size, complexity, and required reliablity. The computer capabilities will remain constant with the microcomputer available to the squadron. Personnel will vary by experience, programming capabilities, language experience, and familiarity with the project under development. The project attributes will vary by software development tools and required development schedule. By assigning multipliers for these attributes, an improved estimate of the development effort is possible. The multipliers are typically plus or minus ten, twenty, or thirty percent, depending on whether the attribute helps or hinders the development. [Ref. 22: pp. 114-126]

The determination of coded program length is determined by the complexity of the project and defined in the requirements stage. Two examples are provided that may serve as base lines for other projects.

Example 1.

The problem objective was to produce a thirty day flight schedule showing assigned crews, flight profile, take-off and landing times, operating area, and recovery base. One aircraft was to remain airborne at all times. Twelve crews, deployed three at a time, would each complete six to ten flights to randomly selected operating areas. Flights would last eight to twelve hours and would recover at randomly selected bases; the last flight for a crew to recover at home field. Figure C.4 shows the overall structure of the design. The program was written with all original code in Pascal and came to 3000 lines. The equations 4.1 and 4.2 yield:

$$MM = 2.4\ (3KDSI)^{1.05} \qquad \text{(eqn 4.3)}$$

$$MM = 7.6 \text{ man-months}$$

$$TDEL = 2.5\ (7.6)^{0.38} \qquad \text{(eqn 4.4)}$$

TDEL = 5.4 months

The program was actually completed in 4.6 man-months by a team of five programmers over a two month period. The shortened development is attributed to familiarity with the problem and the programming language.

Example 2.

The problem, to produce a set of menu selectable driver programs for dBase II, is listed in Appendix B. The program length, as in example 1, is 3000 lines, but the programming tools provided by dBase II and its underlying structure made development easier and faster. Completion time was under two man-months by a single programmer.

A common method for determining the lines of code is to compute an expected value by taking a weighted average of the optimistic a, most likely m, and pessimistic b lines of code [Ref. 23: p. 76]:

$$EXPECTED\ LINES = (a + 4m + b)/6 \qquad \text{(eqn 4.5)}$$

B. ANALYSIS AND DESIGN

A thorough understanding of preferred software design methodology will not only greatly enhance the usefulness of programming efforts within a command, but also reduce future program maintenance and ensure a more reliable product that continues to meet performance expectations. The concepts presented in this section should be understood before command software development projects are approved. Appendix B and Example 1 above, will be used as the development examples throughout this section. The data base program provides personnel information via menu driven screens. After word processing, commands will probably find

data base management programs to be their most useful and time saving applications.

1. Determination of Requirements

Critical to a successful development project is the early, clear definition of what needs to be done. This may be initiated in the form of a request such as "I want a list of all personnel in the squadron who hold a secret clearance" or "I want an updated recall bill." The feasibility of the project should be looked at in terms of:

a) Technology - can the microcomputer system handle the requirements?

b) Economics - will there be sufficient benefits to justify the costs of creating and running the system or are the lost opportunity costs too great to not develop it?

c) Operations - will the system be used or will user resistance prevent full benefits from the system?

With this information, the C.O. can give go-ahead approval. A detailed investigation of specific requirements can then be started. Details about what is to be done, how often, at what volume, by whom, and how well must be completely understood by the developer. [Ref. 21: pp.18-21] The goal of analysis is to produce the optimal solution to the specific business problem under study. The product should be highly maintainable and sized by effective partitioning. Early use of graphical representations for the logical systems will allow the user to evaluate the model and offer criticism and modification suggestions before the implementation phase.

2. Development Phase

The central phase of software development consists of preliminary and detailed design, coding, and testing. Proper design applies various techniques and principles to produce a detailed model that can later be built. The primary tools available are data flow diagrams, data dictionaries, and structure charts.

A key goal of structured analysis is to partition the problem into manageable pieces that can be described in about one page each. The data flow diagram (DFD) is used for this as it provides a graphical representation of the logical model for the system. Figure 4.2 shows the elements of a DFD.

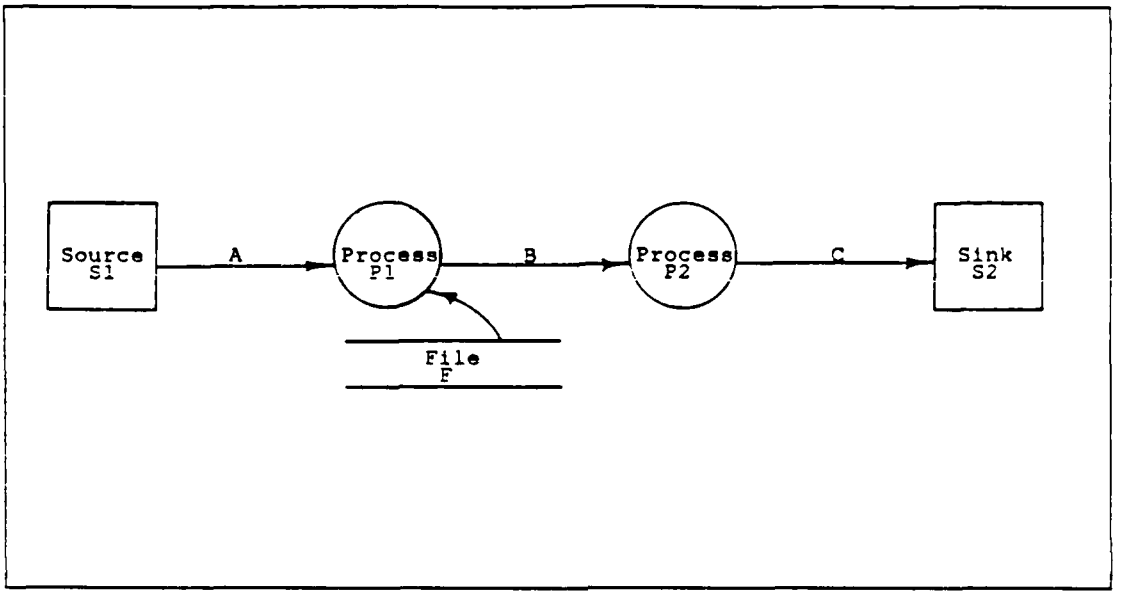


Figure 4.2 Elements of the Data Flow Diagram

It can be seen that A's arrive from the source S1 and are transformed into B's by process P1 (which accesses file F). Process P2 then transforms the B's into C's which are delivered to sink S2. Data flow, represented by the named arrows, is the pipeline through which information flows. Each data flow name should uniquely identify the data. Each process is represented by a bubble, and each process acts on incoming data, transforming it into outgoing data. A process is named with an action phrase that describes the transform in terms of input and output data. The third element of a DFD is a file, shown with a double line. This is a temporary storage area for data, named to describe that

data. A process can access this data for reading or updating. The last element of a DFD is the source or sink, represented by a box. These are people outside the system that originate or receive system data. [Ref. 24: pp. 47-69]

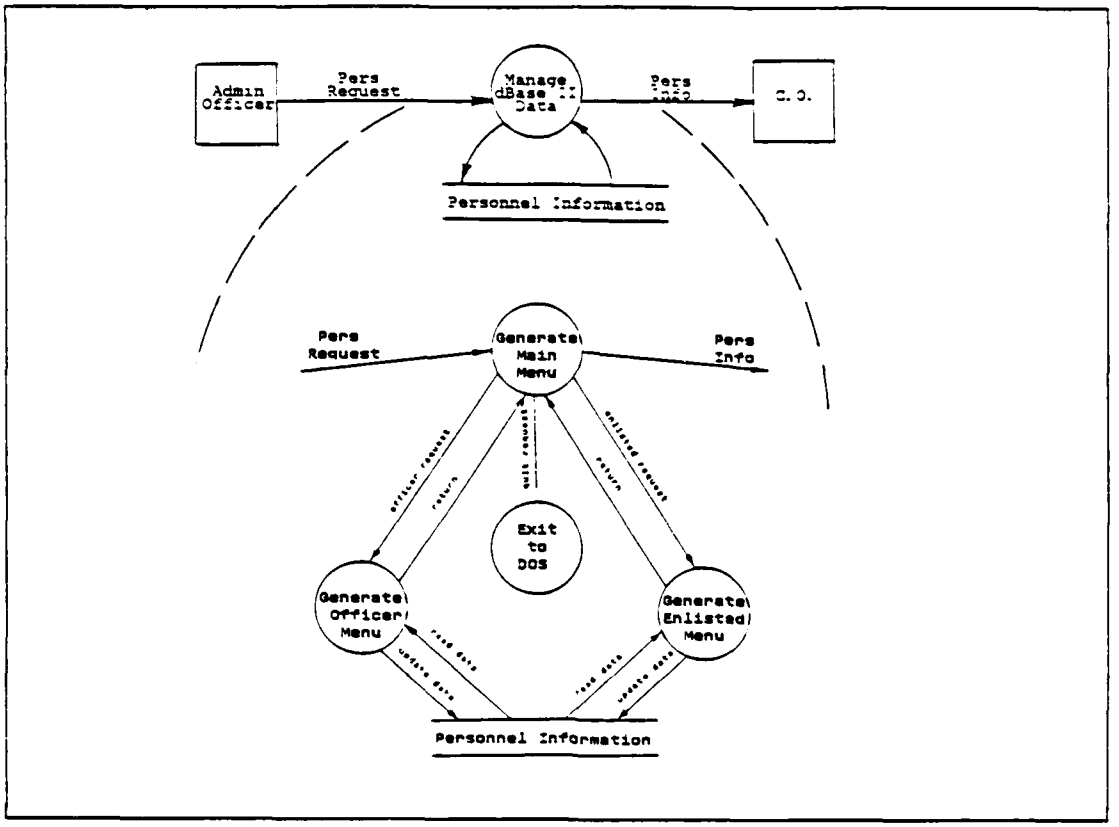


Figure 4.3 Leveled Data Flow Diagram

Development of the full DFD is done in top-down refinement stages. Figure 4.3 shows how Level 0 of the personnel problem is expanded into Level 1 to show more process details. To maintain readability, each level should not be partitioned into more than abut seven elements. If more are needed, a next-lower-level should be created. The leveling is continued until the system is defined in terms of simplified processes that can each be described in a one page specification called a "Mini-Spec." Appendix C shows

how this refinement is carried out for a more complex system.

The DFD provides a picture of the elements and associated flows of data in a system. The Data Dictionary (DD) supports it by fully describing all the elements and flows. In addition it includes the set of procedures used to build and maintain the DD. The DD is a place to look up definitions of the flows, components of flows, files, and processes. The DD and DFD work together to give a full, clearly defined description of the system. A sample DD entry for data flow "BASES AVAILABLE" (J) of Appendix C is shown in Figure 4.4 .

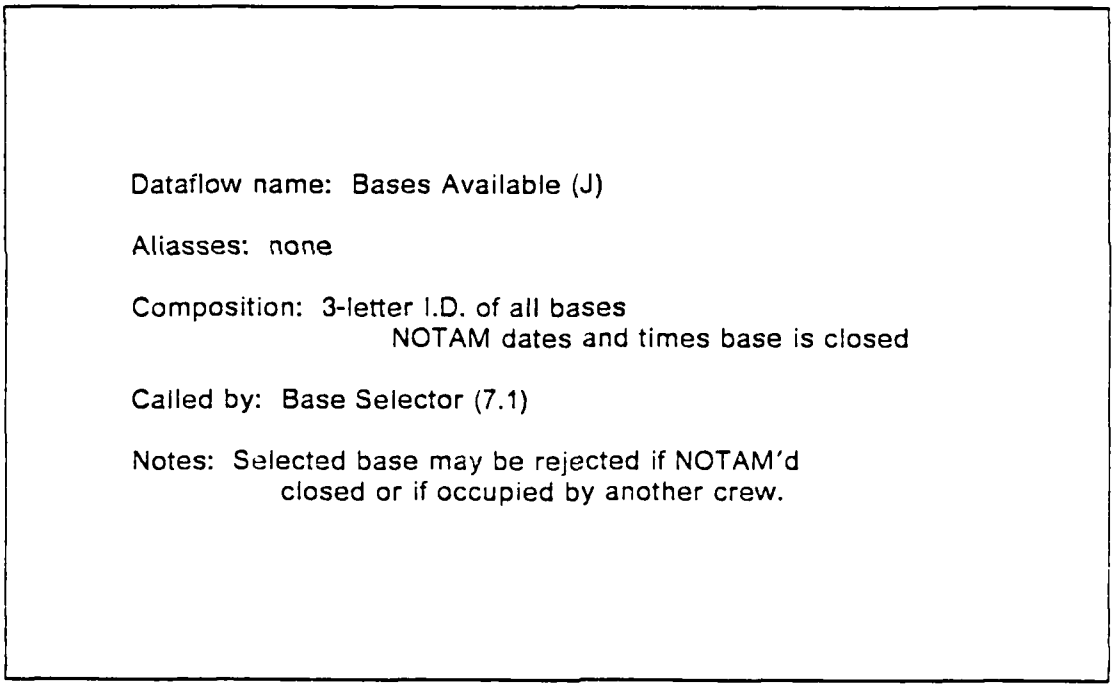


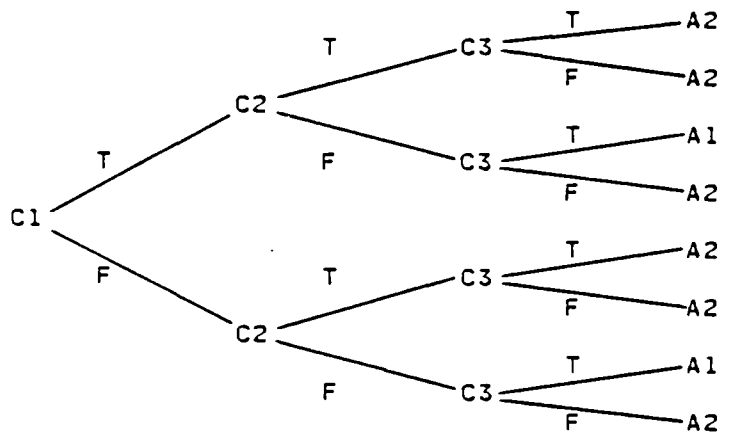
Figure 4.4 Sample Data Dictionary Entry

The DD entry for a process includes a description of how the process will be implemented. The steps of the process are described in structured, English-like "pseudo-code." The elements of the process will be related by one

or more of only three relationships - sequence, decision, or looping. Instead of a text description of the process, a decision tree or decision table can be used to show the process as pictured in Figure 4.5 . The method selected should be the one that can most clearly describe the process. [Ref. 24: pp. 129-178]

Conditions and Actions for CHECK_BASE
(module 16 of Figure C.4)

condition	combinations							
C1	T	T	T	T	F	F	F	F
C2	T	T	F	F	T	T	F	F
C3	T	F	T	F	T	F	T	F
action	decision							
A1			X				X	
A2	X	X		X	X	X		X



conditions	actions
C1 Base occupied	A1 Land at base
C2 Base NOTAM'd closed	A2 Reject and choose alternate base
C3 Last leg of deployment	

Figure 4.5 Decision Tree and Decision Table

After completing the data flow diagram and the data dictionary, the "Structure Chart" may be constructed. The

structure chart is a graphical description of the software modules that will make up the finished system. The chart is constructed by dividing the data flow diagram into three phases of data flow--afferent (inbound), transform (center), and efferent (outbound) flow. The chart is composed of boxes containing the names of the modules. Interconnecting the boxes are lines for data flow. On each line is an arrow showing the direction of flow. The structure chart in Appendix C shows afferent flow from module 1 to module 2, transform flow down the center of the chart, and efferent flow out to module 3. The chart should fan out from the main control module (in this case, module 2) and then fan in again near the lower modules since these common modules are used by more than one module at higher levels in the chart. [Ref. 23: pp. 178-202]

Figure 4.6 illustrates the three steps for converting the data flow diagram into a structure chart. Starting with the inner-most bubble of the afferent flow and working outward, the modules for getting, reading, and making data are charted. These produce "b" for the transform section. The second step begins with a main module which is used to control the transform. It takes in the data just produced, changes it, and outputs to the right. The center modules are constructed by starting at the efferent/transform boundary and moving left to make each module of the transform center. Data "b" is used to make data "c" and data "d", which together make data "e". Data "e" is passed out to the right. The last step is to construct the modules which put or write the data. Here data "e" is formatted into data "f" before being written out. [Ref. 23: pp. 186-192]

The concepts of "cohesion" and "coupling" should be considered while developing the data flow diagrams and structure chart. Cohesion is a measure of the functional

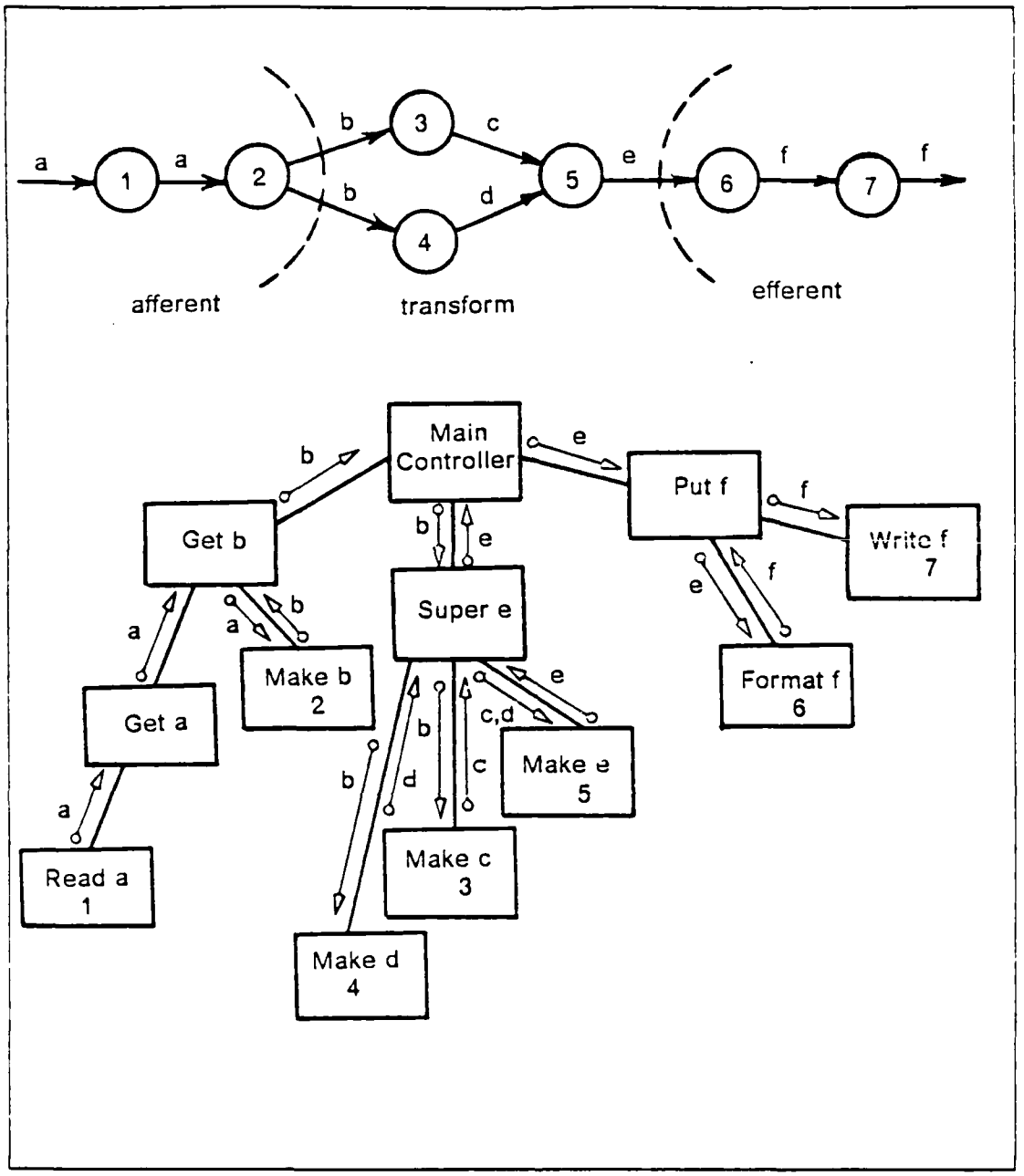


Figure 4.6 Data Flow Diagram and related Structure Chart

strength of a module. A cohesive module does only one task inside a software procedure. If a module contains a group of procedures which must be done in order, it has sequential cohesion. If a module has procedures which must be executed

in the same time span, it is called temporal cohesion. Procedures grouped together without any relationship should be avoided by factoring them out into more modules. Although it is a relative measurement, high cohesion for each module is desired. Coupling measures the interconnection among modules. The degree of coupling should be as low as possible. If an error needs to be corrected in one module, there is less "ripple effect" down through the rest of the modules. Low coupling results in a system that is easier to understand and easier to maintain. [Ref. 23: pp. 158-164]

3. Coding and Testing

Coding follows directly from the structure chart. With good design, efficient code can be written faster and easier. The programmer should strive for code that is simple and direct. The emphasis is on making it work in a straight forward manner, avoiding complicated conditional tests and heavily nested loops. It should be written so that the end user's work is minimized. Prompts for user inputs should be provided, especially if undesireable results could occur like inadvertant erasure of a file or the reformatting of a disk. Each functional section of code should be identified by purpose, author, date, and if changes were added. The code itself should contain "remarks" to explain what each smaller section is doing.

Testing of the coded module can be done in a number of ways. The bottom-up method starts by testing all of the lowest level modules. As these prove good, the next higher level of modules is tested with the lower modules attached. This method requires a "dummy" module at a higher level to initially call the lower modules. The top-down method tests the highest level module first by attaching "stubs" in place of lower modules. The stubs are then replaced with the actual modules and lower stubs are added for more testing

until the system has been "fully tested." The term "fully tested" means "as fully as reasonably possible." A complex program can have thousands of possible data flows, some of which may not be exercised until the system has been put into operation. At that point, good design and thorough documentation are valuable tools in isolating the problem.

4. Phase-in Methods

The method of installing a new system can be a significant factor in its final success. Four strategies exist. The "plunge" method is the most dangerous in that the existing systems (possibly manual systems) are dropped when the new system is started. If a problem developes there is no backup to rely on. A better but more expensive way is to run both systems in "parallel" for a period of time. The "pilot" strategy puts a new system into a small part of the user base, which isolates any damage if the system does not work properly. The last method is to introduce the new system in "phases" into different areas of the organization. [Ref. 8: p. 111] A squadron with a new microcomputer system will probably find itself using each of these methods at certain times. The possibility of failures should be anticipated for any new system.

C. PEOPLE AND COMPUTERS

1. Stress

The split-second accuracy and incredible speed of the computer expands the rapidity and capacity of what can be accomplished - but at a price. Brod [Ref. 25: pp. 16-17] uses the term "technostress" to describe the inability to cope with computer technology in a healthy manner. He breaks this down into technoanxious and technocentered personalities. The technoanxious person feels pressured by his managers, his peers, and the general culture to accept and use computers, but he finds the experience to be painfully slow and uncomfortable. The technocentered person is

highly motivated and eagerly accepts the new technology but exhibits a loss of capacity to relate to others. Insisting on speed and efficiency, he tends to isolate himself, losing patience with those who cannot keep up. His keyboard demands concentration and he developes an intolerance for any interruptions.

Time, for all users, is compressed. The urge to do just one more calculation, or write just one more line of code, can use up a whole morning as one action leads to another and a renewed promise to quit after the next try.

Adapting to the new technology is stressful. The manager must remember that each person, including himself, will adjust and learn to use it at his own pace. Although production may depend on the rate of adaption, the system may never be used properly if it is forced on the user. Workloads must be lowered while the new tasks and procedures are being mastered. Constant monitoring and hovering over the new user will increase his anxieties. The manuals supplied with hardware and software are notoriously poor and make learning a new system even more tedious. A training program that adjusts to different learning rates and capabilities will provide the greatest return for the manager. [Ref. 26: pp. 43-48]

The manager may find that some people view the new system as chance to strengthen their own position through what Brod calls the Star Game, the Ransom Game, and the Maintenance Game [Ref. 25: pp. 59-75].

The Star Game is worked two ways. The player is a clever and aggressive upper level employee who will push for the biggest, fastest, and flashiest new peripheral or software package to keep himself highly visible, but with little planning or implementation forethought. The other variation is the middle manager who senses his supervisor's commitment to the new technology and quickly imposes unreasonable

scheduling and workload demands on his division so he appears to be supportive.

The Ransom Game is the manipulation of time by employees. Traditional work slowdowns were obvious to a manager because be knew what the work entailed and how much time was required. A computer-naive manager will not be able to see this form of protest if he lacks understanding in what the tasks involve. If all he can see and understand is output, he does not know what efforts are required.

The Maintenance Game is a familiar one played in a technical age. One person's knowledge is the key to keeping a particular system running. The person feels extra prestige and will often guard his knowledge to keep his position secure. With a microcomputer, the hardware and commercial software packages should be mastered by a core of people. More importantly, since organizations change, their requirements will change - any specially written programs must be well documented so that maintenance can be performed even after the "expert" leaves.

2. Ergonomics

The comfort and well being of workers can greatly influence their productivity. Too often a new microcomputer is unwrapped and "installed" on the most convenient desk. The National Institute of Occupational Safely and Health (NIOSH) has identified a number of human factors that require special attention by the manager. The two main concerns are eye stress and neck strain.

The video display terminal (VDT) is blamed for burning eyes, double or blurry vision, headaches, and general fatigue. While radiation hazards have been shown to be below normal background levels [Ref. &ws85: p. 35], the VDT still suffers from flickering, screen glare, and poorly formed characters, making it more difficult to read than paper documents. Constant refocusing from the screen to the

draft and back make the eye muscles work harder. If the pressure of a deadline is added, the pupils tend to dilate and make convergence and close focusing even more difficult and therefore contributing to further eyestrain [Ref. 27: p. 6].

A properly designed workstation can minimize these discomforts and lead to improved worker efficiency. The microcomputer screen should be positioned to minimize window and overhead light glare. NIOSH recommends that the screen be in the 10 to 40 degree cone below the user's horizontal eye height and 18 to 28 inches away. The keyboard should be about 23 to 27 inches from the floor; a normal 30-inch height desk is too high for comfortable typing. The manager should remember that people are not all the same size. Adjustable chairs and document holders will lessen back strain and enhance worker comfort and productivity. [Ref. 28: pp. 33,44]

D. SECURITY

An organization which relies on a computer system is highly vulnerable if proper security measures are not in place. The most common type of problem is with software. The stored information on disks is subject to magnetic damage from ringing telephones, printer heads, magnetic document holders, and any other source of even weak magnetic influence. Physical damage to the floppy disks can be caused by bending or paper clipping the floppy's black jacket. The magnetic surface which shows through a small window can easily be scratched and even fingerprints can alter its readability. These problems can largely be eliminated by educating the users.

Less easy to prevent are hardware problems and human error. Although highly reliable, the disk drives and even the micro itself can fail. The hardware system is particularly sensitive to power fluctuations. Aboard ship these

can come with disturbing frequency, whether unplanned load dropping or routine changing of ship power to shore power. People can also cause problems with the hardware and the software. The designer may have overlooked something or the operator may have entered erroneous data. A knowledgeable person may even purposely destroy data or sabotage hardware.

The key to all these problems is an effective recovery method. For a microsystem this means backup copies of programs and data files, and storing them physically away from the original copies. It also means a good training program to reduce errors or oversights. Floppy disks with Privacy Act information must be protected and handled just like printed copies would be. Any classified material produced must be done on TEMPEST secure hardware and the file disks stored in an appropriate safe when not in use. COMNAVAIRPAC [Ref. 29] provides instructions for the security of office information systems and also offers a risk analysis guide for these assets.

V. CURRENT SQUADRON USAGE OF MICROCOMPUTERS

A. MICROCOMPUTER SURVEY

The survey used in this research was intended to gauge current levels of Zenith 120 usage and to identify specific requests for application areas which would improve organizational efficiency. The three part survey was mailed in December 1985 to the twenty-four squadrons under Commander, Anti-Submarine Warfare Wing, Pacific. Eighty percent of the squadrons returned at least one of the forms. Appendix A is the survey.

1. CO/XO Questionnaire

The form used for the Commanding and Executive Officers was intended to provide a top level view of how successfully the microcomputers are being used, what information is desired for decision making, and perceived level of required system knowledge. The following highlights were extracted from the survey:

a) Only two squadrons provide reports related to Admin, the Career Counselor, or the Command Master Chief.

b) Application programs for maintenance and operations are most desired.

c) Two-thirds of present computer reports are not in the desired format.

d) The reports provide only about half of the desired information.

e) Accuracy is increased, but time saved was questionable in one one-third of the responses.

f) Thirty percent of the squadrons had no one holding cognizance over the system. Of those that did, more than half indicated a lieutenant-commander or a lieutenant in charge.

g) Only one squadron had an instruction written on the microsystem.

h) Stress and frustration were caused by too few computers, lack of letter quality printers, and large time investment for training and data entry.

i) Two-thirds felt the supplied software was adequate.

j) Two-thirds had developed specialized programs; only

three squadrons had requested outside assistance.

k) An average of five officers and six to seven of the enlisted in each unit knew how to use the micro, but two squadrons had only one or two people who could use it.

l) Five of the twenty-six C.O.s and X.O.s who responded could personally use the system.

m) Of those who couldn't use the micro, thirteen did not intend to learn, but felt they knew enough to get desired information.

n) Eighty-five percent of the squadrons desired more microcomputer systems.

o) Executive officers tended to know more about the micro than their C.O.s did.

The need for more computers and better quality printers were the two most common requests concerning hardware, although some C.O.s recognized a need for increased RAM to run more sophisticated programs like spelling checkers.

Of the C.O.s and X.O.s who did not plan to learn how to use the micro, all said they did not need to know that much detail to get the information they desired. Of the eight who did plan to learn the system, five felt they did not know enough now to ask the right questions.

2. User Questionnaire

The User Questionnaire was designed to get a more detailed picture of how well the microcomputer had been incorporated as an information system. At the time of the December 1985 survey, the average squadron had owned a microsystem for seven months, although two had owned theirs only one month. The experience level of responding users ranged from none to over three years on various mainframe systems. The following list summarizes user's responses:

a) Most felt it was easy to use, but somewhat slow and limited in memory.

b) The supplied software was considered useful.

c) Satisfaction with system training varied widely, but most stated that more was needed.

d) Mean weekly system usage was twenty-three hours, but with a standard deviation of twelve hours.

e) WordStar was considered to be easy to use and very helpful for general correspondence.

f) Nine squadrons did not use any data base management program.

g) Eight squadrons reported not using any spread sheet program.

h) Only eight squadrons had at least ninety percent of their programs and data backed up.

Hardware problems were noted by squadrons which deploy. The system is difficult to move aboard ship. The CRT packing box is too large to fit through some of the hatches and damage occurs if they are moved without the boxes. The system picks up oil and dirt faster aboard ship, and there is no repair facility for the equipment while deployed. Detachments are not outfitted with a system when deploying and have difficulty gaining access to the ship's system. All record keeping must be done manually, even by those detachments which had been using the parent squadron's microcomputer before deploying.

User comments agreed with the C.O. and X.O. statements that additional computers and letter quality printers are needed.

B. MICROSYSTEM SOFTWARE

1. Programs in Use

The survey identified a number of squadrons which have developed useful programs and applications that could be adapted by other commands with similar needs. The best of these programs are described below.

HSL-31 uses a menu driven dBase II program to keep track of pilot and aircrew qualifications. This program provides a hard copy listing by individual name, the qualification, and its expiration date. It allows on-line updates to the data base with a separate set of menu selections. A similar program keeps track of current flight charts (sectionals, approach plates, SIDs, etc.) and also keeps track of the contents of each navigation bag to ensure

TABLE II

CO QUESTIONNAIRE SUMMARY

Zenith generated reports from:

Admin/CCC/CMC	Yes	08%
OPS	Yes	38%
Training	Yes	50%
NATOPS	Yes	08%
Maint.	Yes	83%

Area which could benefit the most by using the Zenith:
Admin: 08% Ops: 77% Training: 23% NATOPS: 0 Maint: 69%

Computer generated information in the format desired:
Yes: 54% Somewhat: 15% No: 31%

Does it provide the information you need?
Yes: 46% No: 54%

Does the Zenith help the squadron by increasing accuracy?
Yes: 70% Somewhat: 15% No: 15%

Does it always save time?
Yes: 54% No: 46%

Designated person holds cognizance over the system:
Yes: 77% No: 23%

Rate/rank:	LCDR:	40%
	LT to CWO:	44%
	E-7 and below:	16%

Published instruction: Yes: 8%

Indicators of stress or frustration:
-Too few micros
-Time wasted waiting for computer
-No letter quality printer

Adequate software for needs: Yes: 62% No: 38%

Squadron developed any specialized software? Yes: 61%
Areas: Flight proficiency matrix, OPTAR tracking, PME

Outside assistance requested: Yes: 23%
Areas: 3M reporting, Det assignments, Engine tracking

Officers in the squadron who "know" the system:
Mean: 4.3 S.D.: 3.2
Trained in house: All but 1 or 2 per squadron, however some squadrons had no outside training.

Training cycle for teaching the system: Yes: 23%

Do you personally know how to use the Zenith 120 with:

Wordstar	Yes:	15%
dBASE II	Yes:	8%
Lotus 123	Yes:	8%

If no, do you plan to learn any of them: Yes: 23%

CO/XO level needs to know this degree of detail: Yes: 15%

Know enough to ask the right questions: Yes: 77%

Additional microsystem needed: Yes: 92%

TABLE III

XO QUESTIONNAIRE SUMMARY

Zenith generated reports from:

Admin/CCC/CMC	Yes	14%
OPS	Yes	50%
Training	Yes	64%
NATOPS	Yes	21%
Maint.	Yes	50%

Area which could benefit the most by using the Zenith:
Admin: 20% Ops: 40% Training: 47% NATOPS: 7% Maint: 40%

Computer generated information in the format desired:
Yes: 53% No: 47%

Does it provide the information you need?
Yes: 53% No: 47%

Does the Zenith help the squadron by increasing accuracy?
Yes: 86% Somewhat: 7% No: 7%

Does it always save time?
Yes: 53% No: 47%

Designated person holds cognizance over the system:
Yes: 77% No: 23%
Rate/rank: LCDR: 40%
LT to CWO: 44%
E-7 and below: 16%

Published instruction: Yes: 8%

Indicators of stress or frustration:
-Limited programs
-Spead sheet not used to best advantage

Adequate software for needs: Yes: 62% No: 38%

Squadron developed any specialized software? Yes: 61%
Areas: Flight proficiency matrix, OPTAR tracking, PME

Outside assistance requested: Yes: 23%
Areas: 3M reporting, Det assignments, Engine tracking

Officers in the squadron who "know" the system:
Mean: 4.3 S.D.: 3.2
Trained in house: All but 1 or 2 per squadron, however some squadrons had no outside training.

Training cycle for teaching the system: Yes: 23%

Do you personally know how to use the Zenith 120 with:

Wordstar	Yes:	28%
dBASE II	Yes:	8%
Lotus 123	Yes:	28%

If no, do you plan to learn any of them: Yes: 38%

CO/XO level needs to know this degree of detail: Yes: 20%

Know enough to ask the right questions: Yes: 80%

Additional microsystem needed: Yes: 80%

TABLE IV

USER QUESTIONNAIRE SUMMARY

Months squadron has had Zenith 120: Mean: 6.5 S.D.: 3.5

Months personally used system: Mean: 6.5 S.D.: 3.1

Prior computer experience:

None:	24%
Home computer:	38%
Word Prossessor:	19%
Mini:	9%
Mainframe:	9%

Regarding the Zenith system:

	strongly agree		OK	strongly agree		no opinion
Easy to use	32%	21%	47%	-	-	-
fast enough	5%	26%	21%	26%	21%	-
enough memory	16%	26%	16%	21%	21%	-
useful software	32%	26%	42%	11%	-	-
good training by NARDAC	5%	26%	16%	11%	21%	21%

Hours per week system used: Mean: 23 S.D.: 12

Formal method to equitably share the system: Yes: 16%

Regarding Wordstar:
Ease of learning: Easy: 74% Difficult: 26%
do you now feel comfortable using it: Yes: 79%
usefulness: Very: 39%
Limited or cumbersome: 17%
Mixed opinion: 44%

If not used, what word processor is used: Xerox (1 unit)

Regarding Data Base Management programs:
program used: dBase II: 50%
Lotus 123: 5%
None: 45%
what divisions use it: 32% Maintenance 16% Operations
for what: QA analysis, Flight data, rough reports, pilot quals, OPTAR, IMRL, personnel management.

Common data bases used by different divisions: Yes: 5%

Is the DB used in query language only: Yes: 39%

Automated ("one-button") programs used: 21%

Information handled: CAL, IMRL, inventory, Flight data, OPTAR, pilot quals.

Regarding spreadsheet programs:
Program used: Lotus 123: 58%
None: 42%

Used for: 3M summary, End-of-month summary, Flight hours, rosters, pilot/aircrew quals, scheduling.

Percentage of the programs and data "backed-up":
100%: 7 of 19 squadrons
90%: 2 of 19
85%: 2 of 19
80%: 2 of 19
50%: 4 of 19
40%: 1 of 19
20%: 1 of 19

TABLE IV

USER QUESTIONNAIRE (cont'd.)

For VS/HS, difficulties in taking the system aboard ship:
- difficult to move aboard ship
- vibration of moving damages printer
- no repair capability aboard ship
- CRT won't fit through some hatches
- systems get oily and dirty faster

For HC/LAMPS, do the Dets use the ship's system while deployed? Ship's system is different.

Please list any commercial programs the squadron uses which were not supplied with the system package:
None

If you could obtain a custom program for one division, what capabilities would it provide?

- flight time reporting by category, qual, training needed, percent of training achieved.
- Program to tie all operations and maintenance data together.
- Engine efficiency tracking.
- Flight scheduling program.
- Pilot and aircrew flight hours tracking.
- NATOPS information.
- Qualifications tracking (schools, PARS).
- Evaluations.
- Track leave, PRD, EAOS (graph by member)
- Integrated data base/decision support system for flight schedule and flight hour accounting.
- Maintenance training schedule.

no publication is out of date. An auxiliary menu prints a list of squadron officers by lineal number. Lotus 123 is used to keep track of Fleet Replacement Pilot (FRP) and Aircrew (FRAC) training progress. The spread sheet quickly provides a summary sheet of grades and the student's final average for inclusion in his training jacket. WordStar is used to produce the smooth flight schedule. By simplifying the input of new names and events, approximately two man hours per day is being saved.

VS-41 uses a menu driven dBase II program to keep track of squadron OPTAR funds. The easy to use program presents a menu for adding new items, changing present or past quarter items, deleting items, editing allocations, and printing out a finished summary.

HC-11 uses Lotus 123 to keep track of pilots and aircrewmen. The template lists the individual, detachment assignment, and required qualifications with the date completed. The user can then easily sort by individual to show his current qualifications or sort by detachment to indicate that unit's overall readiness level. WordStar files are used to hold listings of schools and convening dates. The appropriate listing is attached to monthly division officer memos to indicate what school openings are available.

HC-1 makes use of dBase II to keep track of H-53 and H-3 high-time components. This is printed on a weekly basis for the maintenance officer for planning aircraft availability. A maintenance personnel listing is used to provide the names of who is attending school, on leave, SIQ, and other manning level information.

HS-4 maintains ASW qualifications on Lotus 123. By making use of the built-in calender function, a monthly listing is generated showing the pilots and aircrewmen and what events they need to requalify in.

HS-2 tracks pilot qualifications using dBase II. Qualification expiration dates are displayed using the index function and then listed by individual. Lotus 123 is used to print end-of-the-month flight time summaries.

HSL-43 maintains detachment personnel and training levels with a menu driven dBase II program. This program can quickly show the readiness level of a selected detachment and indicates which schools or qualifications are still to be completed.

2. Desired Programs

The survey also pointed out a need for application programs in a number of areas. An integrated data base/ decision support system for flight scheduling and flight hour accounting was the most common type of program sought. The need for maintenance department programs was also common. Supply parts tracking, NORS listings, ground training, engine efficiency trends, and quality assurance programs were requested in this area. The need for administrative tracking of personnel on detachment, on leave, their rotation date and EAOS was indicated in the survey.

C. NARDAC SERVICES

The Naval Data Automation Command coordinates a worldwide effort to improve information technology within the Navy. Through twelve regional subcommands, data processing services, hardware and software evaluations, and system development projects are conducted. These regional NARDACs also provide specialized training for other Navy commands. NARDAC, North Island has recently launched a series of classes on microcomputers and specific applications programs. Since NARDAC works under a full charge-back accounting system, typical one-day WordStar or dBase II classes cost the squadron about $95 per day for each student sent. This specialized training provides a solid foundation in the particular software application and permits the

individual to carry this knowledge back to the squadron and in turn train other members of the command.

Although a complete systems analysis and implementation capability exists with the NARDAC, typical development costs can run over $50,000. An extremely economical alternative exists however, since previously developed software is available on disk with full documentation for $100 per application.

VI. CONCLUSIONS AND RECOMMENDATIONS

Information gathered on the current usage of microcomputers by Navy commands indicates that although pockets of effective management exist, most commands do not yet possess the expertise needed to fully utilize their microcomputer systems. Training for individual users is a key factor in improving computer usage, but there is also a need for upper level managers to not only understand the capabilities of the micro, but to also understand how the larger system of the command can integrate these capabilities to enhance the efficiency and effectiveness of the unit and ultimately improve its overall readiness.

A. RECOMMENDATIONS

1. Command Level

The commanding officer should deal with the microcomputer system from an educated, overall viewpoint. He must assess the micro's capabilities in relation to the spectrum of needs within his command and the external demands on the command. Training for his personnel is essential to fully utilize the system. The NARDAC courses offer a good foundation, but more detailed instruction for each application package must be made available. Formal training within the squadron should then transfer this knowledge to more individuals. The training should be geared to the abilities of the individual.

If specialized software is to be developed within a command, the commanding officer should insist on a methodology of established life-cycle considerations. Documentation throughout the process is essential for maintainable products. It also ensures software that can be transported to other similar need areas, both within the

command and to outside users. Those developing the software should not be reinventing the wheel. Commodore Grace Hopper said:

> Why start from scratch with every single program you write? Develop one that (will) do a lot of basic work over and over again for you. [Ref. 30: p.45]

A committee composed of knowledgeable members from each department should be established in the command. This group would evaluate departmental software requests and coordinate the development efforts of this application software. They would investigate outside sources for similar packages, ensure the project has overall benefits for the command, and keep standardization consistent in areas like data structures.

The physical environment for individual users should be made as pleasant as possible. System placement in relation to chair/desk height and lighting conditions needs to be evaluated. Apprehension over the new technology should be addressed.

Computer systems fail. A strict back-up procedure should be in place so that hardware or software problems produce a minimum impact.

2. Training

Adequate training provides the skills needed to gain maximum benefit from the microcomputer system. In addition to the basic and application courses, specialized programming should be made available. This would cover the complete systems analysis approach and thus provide individual commands with "resident experts" who could skillfully guide package development from within house. The programming language for dBase II or dBase III should be offered, as well as the use of prototyping tools like NARDAC's DB2GEN which speeds up program development.

3. Application Programs

The level of success of the microcomputers will depend to a large degree on the sharing among users. User groups and a central software library are recommended. A user group provides a vehicle for expanding computer usage by making the expertise of the group more readily available. It encourages experimentation and exposes the members to more ideas for new applications. It also tends to draw more people in to the knowledgeable user's circle. The Zenith Company runs a users group that may provide useful information and programs. Membership is available from the company at Hilltop Road, Saint Joseph, Missouri, 49085. The Wing and Type Commander levels should maintain a "resident expert" list of people who are especially skilled so that their talent can be used by other commands.

These two echelons should also establish a software library, including full documentation, containing all programs being used in the subordinate commands. Since the Air Force also purchased the same hardware/software package, this library could be enhanced if it included applications from that service as well. The Air Force is presently publishing a listing of certified software being used by their commands.

The following publications would also be a useful part of the library:

- Government Computer News
  GCN Communications Corporation
  1620 Elton Road
  Silver Spring, MD 20903
  (A weekly newspaper format with computer related articles in government and industry.)
- C2MUG Bulletin
  Chief, CECOM
  Attn: AMSEL-FL-SDSD (C2MUG)
  Fort Leavenworth, KS 66027-5600

(A 12-page monthly flyer with hardware and software articles and software catalog, including Zenith.)

- Chips Ahoy
  NAVDAC
  Washington, D. C.
  (A monthly bulletin with Navy related computer information and reviews.)

4. Upgrades

The microcomputer systems supplied to fleet squadrons were part of a standard configuration purchase by the Navy. To improve the usefulness of the systems, the following upgrades are recommended:

a) Increase the RAM available to at least 512K and ideally to 768K.

b) Procure a compatible spelling checker that will work across different word processors, spread sheets, and data base programs.

c) Replace dBase II with dBase III. The newer version can keep more than two index files open, accomodate more and larger files, and runs faster.

d) Letter quality printers should be included in the standard outfitting package.

e) A smart MODEM should also be included.

f) One hard disk drive system should be available to each squadron.

g) A local area network system should be investigated so that all stations can make use of common software and data base information.

This paper is aimed specifically at a squadron commanding officer, but most points are applicable to any manager who must rely on computers to assist him in getting the most out of his assets. Many of these managers received their training and matured in a world without benefit of microcomputers. In recent years the explosive proliferation of micro availability demands that the manager understand their capabilities. If the technology is poorly used, the machine's capabilities are wasted and so are the worker's.

The paper is designed to provide executive level managers with the elementary tools needed to understand how computers work, what makes up a computer system, how to effectively specify application requirements, and how to manage the whole system.

APPENDIX A

MICROCOMPUTER SURVEY QUESTIONNAIRE

The following three part survey was sent to all squadrons under Commander, Anti-Submarine Warfare Wing, U. S. Pacific Fleet. As these units represent the widest variety under a single commander with 6 squadron types and 12 aircraft models, it was felt these squadrons could provide the most accurate small sample look at the current usage of microcomputers in aviation squadrons.

1. COVER LETTER

16 December 1985

From: Cdr M. W. Skahan
SMC 1724
Naval Postgraduate School
Monterey, CA 93943

To: Commanding officer, xxxx

Subj: Zenith 120 Computer Applications

Encl: (1) and (2) CO/XO Questionnaire
(3) User Questionnaire

1. In 1983, in a joint $30 million contract with the U. S. Air Force, the Navy started purchasing the Z-100 series microcomputer in order to equip all commands with a desktop computer. Hardware and limited software was distributed as part of the vasic package, but validated applications software was left to individual users. The enclosed questionnaires are part of a research effort to provide better, more useful software programs and improved methods of managing these microcomputer assets.

2. xxxx's help in filling out the enclosed questionnaires will provide fleet squadrons with more productive application programs and systems management.

3. Please complete enclosure (1) and have your XO complete enclosure (2). The third part of the survey should be completed by the person, officer or enlisted, who you feel has the most knowledge of the Zenith 120 system.

4. Thank you for assisting in this survey. If you would like further information, I can be reached through the CSM office, NPS: (A) 878-2174 or (C) 408-646-2174.

Very respectfully,

M. W. Skahan

2. CO/XO QUESTIONNAIRE

Date:

Please indicate your billet: CO XO

Are you now provided with Zenith generated reports from:

Admin/CCC/CMC	Yes	No
OPS	Yes	No
Training	Yes	No
NATOPS	Yes	No
Maint.	Yes	No

Which area, given the right software, could benefit the most by using the Zenith? _____

Is the computer generated information you receive in the format you desire?

Does it provide the information you need?

Does the Zenith help the squadron by increasing accuracy?

Does it always save time?

Have you designated anyone to hold cognizance over the system? Y N If yes, rate/rank?_____

Do you have a published instruction on the use of the microcomputer system? Y N

What indicators of stress or frustration have you noticed in regard to the system?

Do you think the software provided was adequate for squadron needs? Y N

Has you squadron developed any specialized software? Y N If yes, what areas does it help? _____

Have you requested outside help (Wing, NARDAC, etc.) for assistance in developing special software? Y N If yes, what areas? _____

Approximately how many officers in the squadron "know" the system? ____ and how many enlisted? ____

Of these, how many were trained in house? ____

Do you have a training cycle for teaching the system? Y N

Do you personally know how to use the Zenith 120 with:

Wordstar	Y	N
dBASE II	Y	N
Lotus 123	Y	N

If no, do you plan to learn any of them?

Do you feel CO/XO level needs to know this degree of detail on how the system works?

Do you feel you know enough about the system to ask the right questions so the information you want can be provided?

Would an additional microsystem improve your squadron's productivity?

Other comments, positive or negative, regarding the Zenith 120 microcomputer system:

3. USER QUESTIONNAIRE

Date: Rate/Rank:

Approx date squadron received Zenith 120: _____

Number of months you personally have used system: _____

Prior computer experience

System(s):

Months used:

Fill in the next two lines only if C.O. desires feedback on this survey.

Name:

Organization:

Regarding the Zenith system:

	strongly agree		OK		strongly agree	no opinion
Easy to use	1	2	3	4	5	6
fast enough	1	2	3	4	5	6
enough memory	1	2	3	4	5	6
useful software	1	2	3	4	5	6
good training by NARDAC	1	2	3	4	5	6

How many hours per week is the system used? _____

Is there a formal method used to equitably share the system among all divisions?

Regarding Wordstar:

if used, ease of learning:_____

do you now feel comfortable using it:_____

usefulness:

If not used, what word processor is used?_____

Regarding Data Base Management programs:

program used:

what divisions use it? _____

for what? _____

_____

Are common data bases being used by different divisions? Y N

Is the DB used in query language only? (ie: .list for name = "SMITH") Y N

Does the squadron use any automated programs ("one-button" inputs) to run the database program? Y N

If yes, who wrote it? _____

What information is handled?

Regarding spreadsheet programs:

If used, what program? _____

Division(s) which use it:

What do they use it for?:

Approximately what percentage of the programs and data in the squadron are "backed-up"? _____

For VS/HS, has the squadron experienced any difficulties in taking the system aboard ship? What?

For HC/LAMPS, do the Dets use the ship's system while deployed? Any problems?

Please list any commercial programs the squadron uses which were not supplied with the system package:

If you could obtain a custom program for one division, what capabilities would it provide?

APPENDIX B

SAMPLE DBASE II PERSONNEL MANAGEMENT PROGRAM

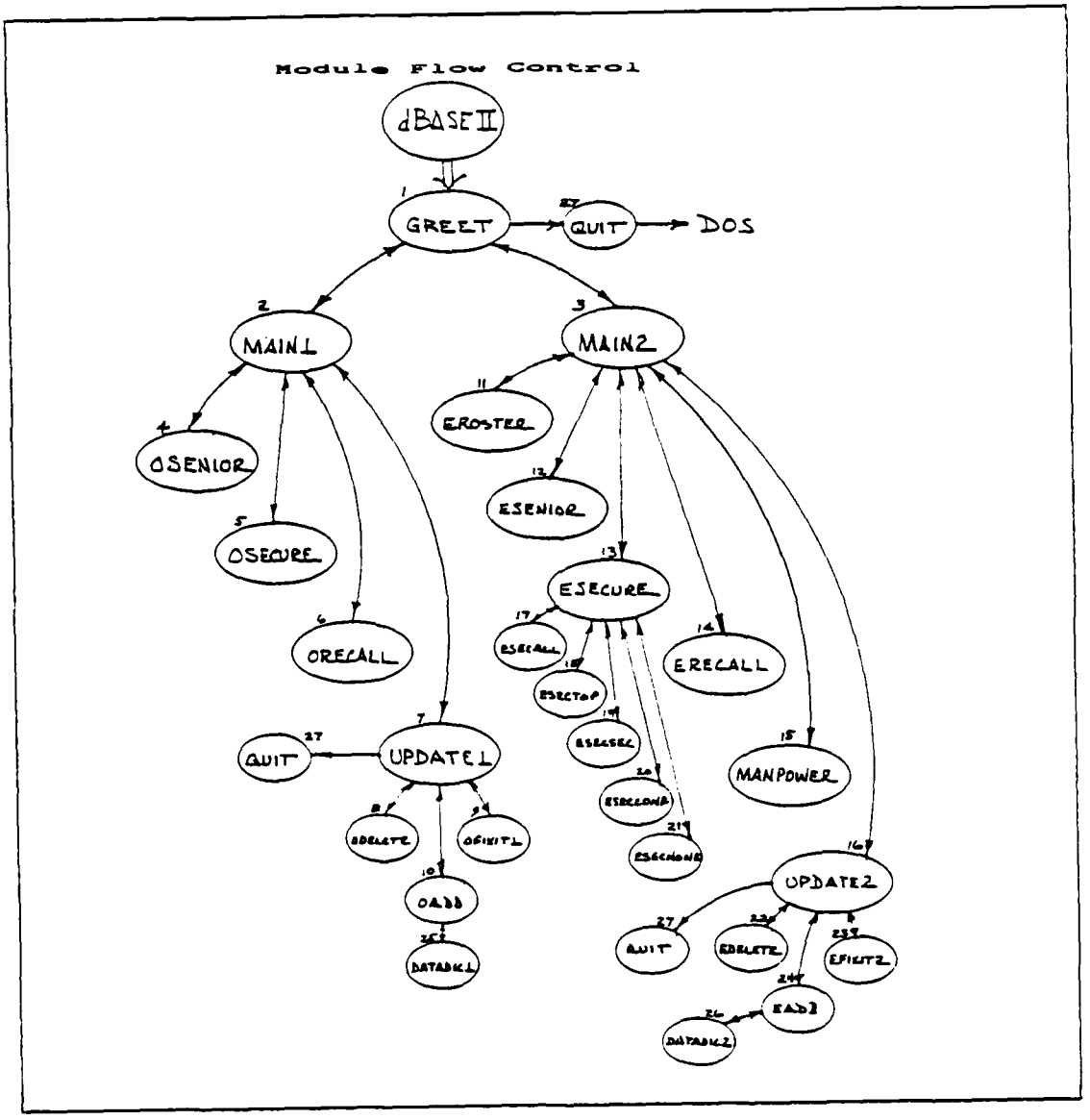


Figure B.1 Module Flow Control

This shows the overall stucture and control of the program. A major decision point is located in Module 1 to separate enlisted from officer data. Also note the available on-line help in Modules 10 and 24 when adding personnel.

1. PROGRAM LISTING

This program may be copied in part or entirely for use as desired.

```
***************                                        *************
***************  Personnel Management System  *************
***************      Written for use on       *************
***************     Zenith 120 computers.     *************
***************                                        *************
**
**  This program was written for use by fleet squadrons
**    which were outfitted with the Zenith 120 computer.
**    Output is to printer.  Adding, deleteing, and changing
**    data does not require a printer.
*
**Module  1 ***  GREET.PRG **************
* GREETING SCREEN
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*  to process enlisted or officer data.
*****************************************
CLEAR
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
@ 1,26 say 'H S  -  85'
@ 2,15 say 'O F F I C E R   &   E N L I S T E D'
* NOTE: Some lines in this program have been split due to
* limits on line length in this listing printout.
* When entered into a micro, they should be typed on
* a single, 80-column line.  The next line is an example.
@ 3,5 say 'P E R S O N N E L   M A N A G E M E N T
   S Y S T E M'
@ 4,27 say 'ver. 1.5'
? " "
? " This program allows additions, deletions, and changes
 to officer"
? "  and enlisted personnel data.  The output is to a
 printer and "
?  "  is menu driven for a total of 12 different forms.
  A help screen"
? "  is available during the 'ADD RECORD' program."
STORE ' :' TO driver
DO WHILE !(driver)<>'A:' .AND. !(driver)<>'B:'
   STORE ' :' TO driver
   @ 16,10 SAY 'PLEASE ENTER THE LETTER  A  THAT
    IDENTIFIES THE'
   @ 18,10 SAY 'DISK DRIVE CONTAINING DATA FILES'
    GET driver PICTURE '!'
   READ
ENDDO
STORE ' ' TO loopit
DO WHILE loopit=' '
   STORE driver+'enlisted' TO enl
```

```
    STORE driver+'officer' TO offr
    IF FILE ('&enl') .AND. FILE ('&offr')
       STORE 'X' TO loopit
       RELEASE enl,offr
    ELSE
       ERASE
       @ 12,10 SAY 'FILES ENLISTED.DBF and OFFICER.DBF
        ARE NOT ALL'
       @ 14,10 SAY 'PRESENT ON THE DRIVE INDICATED.
        PLEASE CHECK.
       @ 20,0 SAY 'PRESS <RESET+CTRL> together to continue.'
       WAIT
    ENDIF
ENDDO

DO WHILE T
ERASE
@   1, 0 SAY "========================================="
@   1,40 SAY "========================================="
@   2, 0 SAY "||"
@   2,16 SAY "HS-85 PERSONNEL INFORMATION SYSTEM"
@   2,78 SAY "||"
@   3, 0 SAY "========================================="
@   3,40 SAY "========================================="
@   4, 0 SAY "||"
@   4,78 SAY "||"
@   5, 0 SAY "||"
@   5,78 SAY "||"
@   6, 0 SAY "||"
@   6,78 SAY "||"
@   7, 0 SAY "||"
@   7,78 SAY "||"
@   8, 0 SAY "||"
@   8,78 SAY "||"
@   9, 0 SAY "||"
@   9,78 SAY "||"
@  10, 0 SAY "||"
@  10,78 SAY "||"
@  11, 0 SAY "||"
@  11,78 SAY "||"
@  12, 0 SAY "||"
@  12,78 SAY "||"
@  13, 0 SAY "||"
@  13,78 SAY "||"
@  14, 0 SAY "========================================="
@  14,40 SAY "=====================================mws=="
@   5,20 SAY " 0. QUIT to DOS "
@   6,10 SAY "1. ENLISTED PROGRAMS  2. OFFICER PROGRAMS"
@   7,10 SAY "     Squadron Roster       List by Ranks"
@   8,10 SAY "     Seniority list        Recall Bill"
@   9,10 SAY "     Security Clearan      Security Clearances"
@  10,10 SAY "     Recall Bill            UPDATE data base"
@  11,10 SAY "     Manpower Listing"
@  12,10 SAY "     UPDATE data base"
STORE 3 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum >  2
    STORE " " TO select
   *@ 13,0 say "||                                       "
    @ 13,33 SAY " select (0-2) : : "
    @ 13,48 GET select PICTURE "#"
    READ
    STORE VAL(select) TO selectnum
ENDDO

DO CASE
    CASE selectnum= 0
       SET COLON ON
       SET BELL ON
       SET TALK ON
       CLEAR
```

```
      DO QUIT
   CASE selectnum= 1
   *  DO the enlisted programs
      DO MAIN2
   CASE selectnum= 2
   *  DO the officer programs
      DO MAIN1
   OTHERWISE
      erase
     *@ 7,10 SAY CHR(7) + "INVALID ANSWER, USE 0, 1, or 2."
ENDCASE

STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: GREET.PRG

***************  MAIN1.PRG *************
* MAIN MENU for OFFICER PROGRAMS
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*  to process officer data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF

ERASE
* draw the screen with choices
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,12 SAY "HS-85 OFFICER PERSONNEL INFORMATION SYSTEM"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "=========================================="
@ 13,40 SAY "=========================================="
@  5,20 SAY " 0. exit to ENLISTED Programs or QUIT"
@  6,20 SAY " 1. List by Ranks "
@  7,20 SAY " 2. Security Clearances"
@  8,20 SAY " 3. Recall Bill"
@  9,20 SAY " 4. UPDATE data base"
STORE 5 TO selectnum

* get & process the selection
DO WHILE selectnum < 0 .OR. selectnum >  4
   STORE " " TO select
```

```
   @ 12,33 SAY " select (0-4) : : "
   @ 12,48 GET select PICTURE "#"
   READ
   STORE VAL(select) TO selectnum
ENDDO

DO CASE
   CASE selectnum= 0
     *GO BACK TO GREETING MENU
      ERASE
      RETURN
   CASE selectnum= 1
   *  DO list by ranks
      DO OSENIOR
   CASE selectnum= 2
   *  DO security listing
      DO OSECURE
   CASE selectnum= 3
   *  DO officer recall bill
      DO ORECALL
   CASE selectnum= 4
   *  DO the additions or updating
      DO UPDATE1
   OTHERWISE
      erase
      @ 7,10 SAY CHR(7) + "INVALID ANSWER, TRY 0 - 4"
ENDCASE

STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: MAIN1.PRG

*************** MAIN2.PRG **************
* MAIN MENU for ENLISTED PROGRAMS
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*  to process enlisted data.
****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@  2, 0 SAY "||"
@  2,12 SAY "HS-85 ENLISTED PERSONNEL INFORMATION SYSTEM"
@  2,78 SAY "||"
@  3, 0 SAY "========================================="
@  3,40 SAY "========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
```

```
@ 12, 0 SAY "||"
@@ 12,78 SAY "||"
@@@ 13, 0 SAY "=========================================="
@@@@ 13,40 SAY "=========================================="
@@@@@ 5,20 SAY " 0. exit to OFFICER Programs or QUIT"
@@@@@@ 6,20 SAY " 1. Squadron Roster"
@@@@@@@ 7,20 SAY " 2. Seniority list"
@@@@@@@@ 8,20 SAY " 3. Security Clearances"
@@@@@@@@@ 9,20 SAY " 4. Recall Bill"
@@@@@@@@@@ 10,20 SAY " 5. Manpower Listing"
@@@@@@@@@@@ 11,20 SAY " 6. UPDATE data base"
STORE  7 TO selectnum
*set echo on
 *          set step on
  *                    set talk on

DO WHILE selectnum < 0 .OR. selectnum >  6
   STORE " " TO select
   @ 12,33 SAY " select (0-6) : "
   @ 12,48 GET select PICTURE "#"
   READ
   STORE VAL(select) TO selectnum
ENDDO

DO CASE
   CASE selectnum= 0
     *SET COLON ON
     *SET BELL ON
     *SET TALK ON
      erase
      CLEAR
      RETURN
   CASE selectnum= 1
   *  DO the squadron roster
      DO EROSTER
   CASE selectnum= 2
   *  DO enlisted seniority listing E-9 to E-1
      DO ESENIOR
   CASE selectnum= 3
   *  DO enlisted security listing
      DO ESECURE
   CASE selectnum= 4
   *  DO enlisted recall bill
      DO ERECALL
   CASE selectnum= 5
   *  DO rotation date and loss date, etc.
      DO MANPOWER
   CASE selectnum= 6
   *  DO the additions or updating
      DO UPDATE2
   OTHERWISE
      ERASE
      @ 7,10 SAY CHR(7) + "INVALID ANSWER, TRY 0 - 6"
ENDCASE
?
?
STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: MAIN2.PRG

*************  OSENIOR.PRG *************
* Officer Seniority Listing
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  a listing of Officer by Rank
*****************************************
SET TALK OFF
```

```
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,20 SAY "HS-85 OFFICERS by RANKS"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a Ranking list"
@  6,5 SAY "  for officers in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "Any other key will return you to OFFICER Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
? '                                                  ',DATE()
? '              HS-85 OFFICERS by RANKS'
? ' '
? ' Rank         Name                          SSN'
?
SET PRINT OFF

ERASE
@ 5,0 SAY ' Now getting officer ranks...'

  STORE '                          ' to Zname
  STORE '        ' to Zrank
  STORE '                  to Zssn
   STORE driver+'officer' TO drivefile
   USE &drivefile
```

```
 index on name to &lname
 use &drivefile index &lname

 GOTO TOP
 STORE 0 TO cnt
DO WHILE .not. EOF
 STORE name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'CDR'
    SET PRINT ON
    ? Zrank,' ',Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
   SET PRINT ON
      ?  '    No Commanders in Squadron'
   SET PRINT OFF
ENDIF

 GOTO TOP
 STORE 0 TO cnt
 SET PRINT ON
    ?
 SET PRINT OFF
DO WHILE .not. EOF
 STORE name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'LCDR'
    SET PRINT ON
    ? Zrank,' ',Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
   SET PRINT ON
      ?  '    No LCDRs in Squadron.'
   SET PRINT OFF
ENDIF

 GOTO TOP
 STORE 0 TO cnt
 SET PRINT ON
    ?
 SET PRINT OFF
DO WHILE .not. EOF
 STORE name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'LT'
    SET PRINT ON
    ? Zrank,' ',Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
   SET PRINT ON
      ?  '    No Lieutenants in Squadron'
   SET PRINT OFF
```

```
ENDIF

 GOTO TOP
 STORE 0 TO cnt
 SET PRINT ON
    ?
 SET PRINT OFF
DO WHILE .not. EOF
 STOR name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'LTJG'
    SET PRINT ON
    ? Zrank,    ,Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
   SET PRINT ON
      ?    No Lieutenants, Junior Grade in Squadron'
   SET PRINT OFF
ENDIF


 GOTO TOP
 STORE 0 TO cnt
 SET PRINT ON
    ?
 SET PRINT OFF
DO WHILE .not. EOF
 STORE name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'ENS'
    SET PRINT ON
    ? Zrank,    ,Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
   SET PRINT ON
      ?    No Ensigns in Squadron'
   SET PRINT OFF
ENDIF

 GOTO TOP
 STORE 0 TO cnt
 SET PRINT ON
    ?
 SET PRINT OFF
DO WHILE .not. EOF
 STORE name TO Zname
 STORE rank TO Zrank
 STORE ssn TO Zssn
 STORE TRIM(!(Zrank)) TO Zrank
 IF Zrank = 'CWO'
    SET PRINT ON
    ? Zrank,    ,Zname,' ',Zssn
    SET PRINT OFF
    STORE 1 TO cnt
 ENDIF
 SKIP
ENDDO
IF cnt = 0
```

```
      SET PRINT ON
         ?    No Warrant Officers in Squadron'
      SET PRINT OFF
   ENDIF

SET PRINT ON
   ?
   EJECT
SET PRINT OFF
    *to clear buffer
RELEASE answer,Zname,Zrank,Zssn,cnt
RETURN
ENDDO
* EOF: OSENIOR.PRG

***************  OSECURE.PRG **************
* Officer Security Listing
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  a listing of Officer clearances
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,20 SAY "HS-85 OFFICER CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a clearance listing"
@  6,5 SAY "  for officers in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you
 to OFFICER Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer)<>"P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
```

```
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
? '                                                    ',DATE()
? '                HS-85 OFFICER CLEARANCES'
? ' '
? ' Rank       Name                          SSN
  'Clearance'
? '
SET PRINT OFF


ERASE
@ 5,0 SAY ' Now getting officer clearances...'

  STORE '                          ' to Zname
  STORE '        ' to Zrank
  STORE '                to Zssn
  STORE '                          ' to Zsecclr
   STORE driver+'officer' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   GOTO TOP
   STORE 0 TO cnt
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rank TO Zrank
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'TOP SECRET'
      SET PRINT ON
      ? Zrank,'  ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
      STORE 1 TO cnt
   ENDIF
   SKIP
  ENDDO
  IF cnt = 0
     SET PRINT ON
        ?    No Top Secret Clearances'
     SET PRINT OFF
  ENDIF

   GOTO TOP
   STORE 0 TO cnt
   SET PRINT ON
      ?
   SET PRINT OFF
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rank TO Zrank
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'SECRET'
      SET PRINT ON
      ? Zrank,'  ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
      STORE 1 TO cnt
   ENDIF
```

```
   SKIP
  ENDDO
  IF cnt = 0
     SET PRINT ON
        ?'     No Secret Clearances'
     SET PRINT OFF
  ENDIF

   GOTO TOP
   STORE 0 TO cnt
   SET PRINT ON
      ?
   SET PRINT OFF
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rank TO Zrank
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'CONFIDENTIAL'
      SET PRINT ON
      ? Zrank,'  ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
      STORE 1 TO cnt
   ENDIF
   SKIP
  ENDDO
  IF cnt = 0
     SET PRINT ON
        ?'     No Confidential Clearances'
     SET PRINT OFF
  ENDIF

   GOTO TOP
   STORE 0 TO cnt
   SET PRINT ON
      ?
   SET PRINT OFF
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rank TO Zrank
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'NONE'
      SET PRINT ON
      ? Zrank,'  ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
      STORE 1 TO cnt
   ENDIF
   SKIP
  ENDDO
  IF cnt = 0
     SET PRINT ON
        ?'     Everyone has a clearance.'
     SET PRINT OFF
  ENDIF

SET PRINT ON
   ?
   EJECT
SET PRINT OFF
    *to clear buffer
RELEASE answer,Zname,Zrank,Zssn,Zsecclr,cnt
RETURN
ENDDO
* EOF: OSECURE.PRG

***************  ORECALL.PRG **************
* Officer Recall Bill
```

```
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  an Officer Recall Bill.
****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,15 SAY "    HS-85 OFFICER RECALL BILL"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a RECALL BILL"
@  6,5 SAY "  for all Officers in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "Any other key will return you to OFFICER Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
@ 5,0 SAY ' Now putting OFFICERS in alphabetical order...'
STORE driver+'officer' to drivefile
USE &drivefile
INDEX ON name to &LNAME
USE &drivefile INDEX &lname
@ 10,0 SAY ' Now transferring to printer.'
REPORT FORM orecall TO PRINT
RELEASE answer
RETURN
ENDDO

* EOF: ORECALL.PRG
```

```
**************  UPDATE1.PRG *************
* UPDATE MENU for OFFICER PROGRAMS
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*  to update officer data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF

ERASE
* draw the screen with choices
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,12 SAY "HS-85 OFFICER DATA BASE UPDATE SELECTIONS"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "=========================================="
@ 13,40 SAY "=========================================="
@  5,20 SAY " 0. exit to OFFICER Programs "
@  6,20 SAY " 1. ADD a new oficer"
@  7,20 SAY " 2. DELETE an officer"
@  8,20 SAY " 3. CHANGE an existing officer's data"
STORE 4 TO selectnum

* get & process the selection
DO WHILE selectnum < 0 .OR. selectnum >  3
   STORE " " TO select
   @ 12,33 SAY " select (0-3) : : "
   @ 12,48 GET select PICTURE "#"
   READ
   STORE VAL(select) TO selectnum
ENDDO

DO CASE
   CASE selectnum= 0
   *  Go back to OFFICER MENU
      ERASE
      CLEAR
      RETURN
   CASE selectnum= 1
   *  Do the addition of a member
      DO OADD
   CASE selectnum= 2
   *  Do the deletion of a member
      DO ODELETE
   CASE selectnum= 3
```

```
   *  Do a change to the data
      DO OFIXIT
   OTHERWISE
      erase
      @ 7,10 SAY CHR(7) + "INVALID ANSWER, TRY 0 - 3"
ENDCASE

STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: UPDATE1.PRG (officers)

***************  OADD.PRG **************
* ADD OFFICER DATA
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to add
*   officer data.
****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
store y to addmore
ERASE
@ 1,26 say 'H S  -  85'
@ 2,22 say 'A D D   D A T A   TO  '
@ 3,15 say 'O F F I C E R   P E R S O N N E L'
STORE 'N' TO YESNO
@ 6,8 SAY "Press the Y key to see
 an explanation of the data fields."
@ 7,9 SAY "Otherwise just hit the space bar."
@ 8,8 GET YESNO
READ
IF !(YESNO) = 'Y'
   DO DATADIC1
ENDIF

DO WHILE addmore
ERASE
STORE driver+'odummy' TO drivefile
USE &drivefile
DELETE ALL
PACK
APPEND BLANK

@ 1,0 SAY '        HS-85'
@ 2,0 SAY 'ADD NEW OFFICER DATA'
@ 4,0 SAY '1. LAST NAME, FIRST NAME, MI ' GET name;
 PICTURE '!!!!!!!!!!!!!!!!!!!!!'
@ 5,0 SAY '2. RANK ' GET rank PICTURE '!!!!!!!'
@ 6,0 SAY '3. SSN ' GET ssn PICTURE '!!!!!!!!!!!'
@ 7,0 SAY '4. STREET ' GET street
 PICTURE '!!!!!!!!!!!!!!!!!!!!'
@ 8,0 SAY '5. CITY ' GET city PICTURE '!!!!!!!!!!!!!!!!'
@ 9,0 SAY '6. STATE ' GET state PICTURE '!!!!!'
@ 10,0 SAY '7. ZIP ' GET zip PICTURE '!!!!!!!!!'
@ 11,0 SAY '8. HOME PHONE ' GET hphone
 PICTURE '!!!!!!!!!!!!!!'
@ 12,0 SAY '9. BUSINESS PHONE ' GET bphone
 PICTURE '!!!!!!!!!!!!!!'
@ 13,0 SAY '10. SECURITY CLEARANCE ',
   GET seccl PICTURE '!!!!!!!!!!!!!!!!!!!'
@ 14,0 SAY '11. GEO. ZONE ' GET zone PICTURE '!!!!'

STORE ' ' TO loopit
DO WHILE loopit=' '
STORE ' ' TO decide
@ 23,25 SAY 'MAIN MENU = M, CANCEL = C,
 PROCESS = P' GET decide PICTURE '!'
```

```
READ
IF !(decide)='M' .OR. !(decide)='C' .OR. !(decide)='P'
  STORE '!' TO loopit
ENDIF
ENDDO

IF !(decide) = 'P'
  STORE driver+'officer' TO drivefile
  USE &drivefile
  STORE driver+'odummy' TO addum
  APPEND FROM &addum.
  @ 22,45 SAY 'FUNCTION COMPLETED'
  STORE ' ' TO loopit

  DO WHILE loopit=' '
   STORE ' ' TO decide
   @ 23,25 SAY 'MAIN MENU = M, CANCEL = C, PROCESS = P ';
      GET decide PICTURE '!'
   READ
   IF !(decide)='M' .OR. !(decide)='C' .OR. !(decide)='P'
     STORE 'X' TO loopit
   ENDIF
  ENDDO
   IF !(decide) = 'P'
         STORE Y TO addmore
   ELSE
         STORE N TO addmore
   ENDIF decide
ELSE
    STORE N TO addmore
ENDIF decide
ENDDO addmore

RELEASE addum,decide,loopit,addmore
USE
  STORE driver+'officer' TO drivefile
RETURN
* EOF: OADD.PRG

***************  ODELETE.PRG **************
* DELETE OFFICER RECORDS
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to delete
*  officer records from data base..
*****************************************
SET TALK OFF
SET BELL OFF
ERASE
STORE Y TO dropum
DO WHILE dropum
   ERASE
   STORE driver+'officer' TO drivefile
   USE &drivefile
   INDEX on name to &lname
    USE &drivefile index &lname

   @ 1,0 SAY '         HS-85'
   @ 2,0 SAY 'DELETE OFFICER FROM RECORDS'

   STORE '                          ' TO member
   STORE '                          ' TO Zname
   STORE '        ' TO Zrank
   STORE '                TO Zssn
   STORE '                          ' TO Zstreet
   STORE '                    ' TO Zcity
   STORE '      ' TO Zstate
   STORE '            TO Zzip
   STORE '                ' TO Zhphone
   STORE '                ' TO Zbphone
```

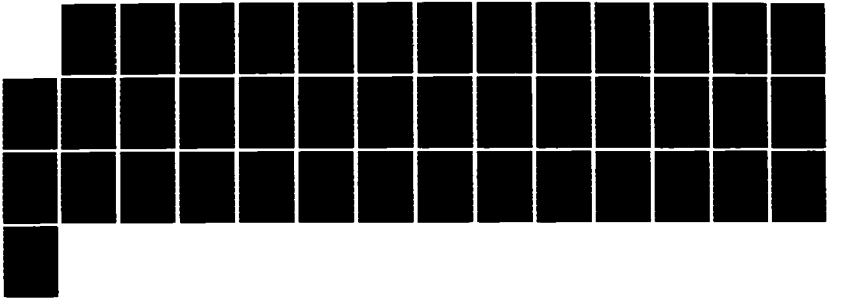
```
    STORE '                              ' TO Zsecclr
    STORE '     ' TO Zzone

      @ 6,22 say "To EXIT, push ENTER only."
      @ 4,0 SAY "Who do you want to DELETE ?";
        GET member PICTURE '!!!!!!!!!!!!!!!!!!!!'
      READ
      IF member = ' '
         RETURN
      ENDIF
    STORE TRIM(!(member)) to name
    FIND &NAME

    STORE # to recnum
erase
  IF recnum = 0
    @ 10,20 SAY 'That spelling NOT found,'
    @ 12,15 SAY 'try again.
      A short version like  SM¦  for SMITH'
    @ 14,15 SAY 'is OK to use.
    @ 18,5 SAY 'HIT ANY KEY to continue.'
    set console OFF
      WAIT
    set console ON
  ELSE

    STORE name TO Zname
    STORE rank TO Zrank
    STORE ssn TO Zssn
    STORE street TO Zstreet
    STORE city TO Zcity
    STORE state TO Zstate
    STORE zip TO Zzip
    STORE hphone TO Zhphone
    STORE bphone TO Zbphone
    STORE secclr TO Zsecclr
    STORE zone TO Zzone
     @ 1,0 SAY '                HS-85   DELETE PERSONNEL DATA'
     @ 3,0 SAY '     This will DELETE,
      this person from the records. '

    @ 6,0 SAY '1. NAME '
      @ 6,9 SAY Zname
    @ 7,0 SAY '2. RANK '
      @ 7,9 SAY Zrank
    @ 8,0 SAY '3. SSN '
      @ 8,9 SAY Zssn
    @ 9,0 SAY '4. STREET '
      @ 9,11 SAY Zstreet
    @ 10,0 SAY '5. CITY '
      @ 10,11 SAY Zcity
    @ 11,0 SAY '6. STATE '
      @ 11,11 SAY Zstate
    @ 12,0 SAY '7. ZIP '
      @ 12,11 SAY Zzip
    @ 13,0 SAY '8. HOME PHONE '
      @ 13,19 SAY Zhphone
    @ 14,0 SAY '9. BUSINESS PHONE '
      @ 14,19 SAY Zbphone
    @ 15,0 SAY '10. SECURITY CLEARANCE '
      @ 15,15 SAY Zsecclr
    @ 16,0 SAY '11. GEO. ZONE '
      @ 16,15 SAY Zzone
  @ 22,33 SAY 'To DELETE this person type "D"'
  STORE ' ' TO loopit
  DO WHILE loopit=' '
     STORE ' ' TO decide
     @ 23,20 SAY 'OFFICER MENU = M, CANCEL = C, DELETE = D'
     @ 23,63 GET decide
     READ
```

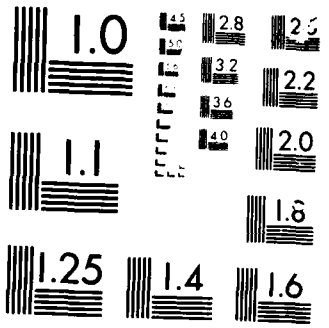
```
      IF !(decide)='M' .OR. !(decide)='C' .OR. !(decide)='D'
        STORE 'X' TO loopit
      ENDIF
   ENDDO

   IF !(decide) = 'D'
       DELETE
       PACK
       STORE ' ' TO decide
       @ 22,12 SAY 'FUNCTION COMPLETED:'
       STORE ' ' TO loopit
       DO WHILE loopit=' '
          STORE ' ' TO decide
           @ 23,20 SAY 'UP DATE MENU = M,
            CANCEL = C, DO MORE = D'
           @ 23,63 GET decide
           READ
          IF !(decide)='M' .OR. !(decide)='C'
           .OR. !(decide)='D'
             STORE 'X' TO loopit
          ENDIF
       ENDDO
       IF !(decide) = 'D'
                 STORE Y TO dropum
       ELSE
                 STORE N TO dropum
       ENDIF decide
    ELSE
       STORE N TO dropum
    ENDIF decide
   USE
ENDDO dropum
USE
RELEASE dropum,member,loopit,decide
RETURN
* EOF: ODELETE.PRG

**************  OFIXIT1.PRG **************
* FIX OFFICER DATA
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to change values in
*   officer data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
STORE 0 TO recnum
DO WHILE t
   ERASE
@ 1,0 SAY ' Getting OFFICER DATA....'
   STORE driver+'officer' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   @ 1,0 SAY '      HS-85                '
   @ 2,0 SAY 'CHANGE OFFICER DATA'
   STORE '                          ' TO member
   STORE '                          ' TO Zname
   STORE '        ' TO Zrank
   STORE '                TO Zssn
 * STORE '       ' TO Zdor
   STORE '                          ' TO Zstreet
   STORE '                     ' TO Zcity
   STORE '      ' TO Zstate
   STORE '            TO Zzip
   STORE '               ' TO Zhphone
   STORE '               ' TO Zbphone
```

```
   STORE '                              ' TO Zsecclr
   STORE '     ' TO Zzone

     @ 6,22 say "To EXIT, push ENTER only."
     @ 4,0 SAY "Who's data needs update?";
       GET member PICTURE '!!!!!!!!!!!!!!!!!!!!'
     READ
     IF member = ' '
       RETURN
     ENDIF
   STORE TRIM(!(member)) to name
   FIND &NAME

   STORE # to recnum
erase
  IF recnum = 0
    @ 10,3 SAY '    That spelling NOT found,
     a shortened version'
    @ 12,3 SAY 'like  SMI  for SMITH is acceptable.'
    @ 15,3 SAY 'To try again, Push any key.'
    set console OFF
      WAIT
    set console ON
  ELSE

   STORE name TO Zname
   STORE rank TO Zrank
   STORE ssn TO Zssn
 * STORE dor TO Zdor
   STORE street TO Zstreet
   STORE city TO Zcity
   STORE state TO Zstate
   STORE zip TO Zzip
   STORE hphone TO Zhphone
   STORE bphone TO Zbphone
   STORE secclr TO Zsecclr
   STORE zone TO Zzone
    @ 1,0 SAY '                HS-85 CHANGE OFFICER DATA'
    @ 2,0 SAY '              If entry is good, hit ENTER.'
    @ 3,0 SAY '     To make CHANGE,
     Type over old entries and ENTER.'
@ 6,0 SAY '1. LAST NAME  FIRST NAME  MI ' SAY Zname
  @ 6,29 get Zname PICTURE '!!!!!!!!!!!!!!!!!!!!'

@ 7,0 SAY '2. RANK ' SAY Zrank
  @ 7,9 get Zrank PICTURE '!!!!!!!'

@ 8,0 SAY '3. SSN ' SAY Zssn
  @ 8,7 get Zssn PICTURE '!!!!!!!!!!!!'

 *dor etc.

@ 9,0 SAY '4. STREET ' SAY Zstreet
  @ 9,11 get Zstreet PICTURE '!!!!!!!!!!!!!!!!!!!!'

@ 10,0 SAY '5. CITY ' SAY Zcity
  @ 10,9 get Zcity PICTURE '!!!!!!!!!!!!!!!'

@ 11,0 SAY '6. STATE ' SAY Zstate
  @ 11,10 get Zstate PICTURE '!!!!!'

@ 12,0 SAY '7. ZIP ' SAY Zzip
  @ 12,8 get Zzip PICTURE '!!!!!!!!!'

@ 13,0 SAY '8. HOME PHONE ' SAY Zhphone
  @ 13,15 get Zhphone PICTURE '!!!!!!!!!!!!!'

@ 14,0 SAY '9. BUSINESS PHONE ' SAY Zbphone
  @ 14,19 get Zbphone PICTURE '!!!!!!!!!!!!!'
```

AD-A169 263 EFFECTIVE MICROCOMPUTER MANAGEMENT: AN EXECUTIVE LEVEL GUIDE(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M W SKAHAN 27 MAR 86 2/2

UNCLASSIFIED F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

```
@ 15,0 SAY '10. SECURITY CLEARANCE ' SAY Zsecclr
  @ 15,23 get Zsecclr PICTURE '!!!!!!!!!!!!!!!!!!!!'

@ 16,0 SAY '11. GEO. ZONE ' SAY Zzone
  @ 16,14 get Zzone PICTURE '!!!!'
READ
IF Zname = '                              '
   RETURN
ENDIF
ERASE
   @ 2,0 SAY '  Your new changes look like this '
   @ 6,0 SAY '1. LAST NAME  FIRST NAME  MI '
     @ 6,31 SAY Zname
   @ 7,0 SAY '2. RANK '
     @ 7,31 SAY Zrank
   @ 8,0 SAY '3. SSN '
     @ 8,31 SAY Zssn
  *@ 11,30 SAY '0. DATE of RANK '
   * @ 11,19 SAY Zdor
   @ 9,0 SAY '4. STREET '
     @ 9,31 SAY Zstreet
   @ 10,0 SAY '5. CITY '
     @ 10,31 SAY Zcity
   @ 11,0 SAY '6. STATE '
     @ 11,31 SAY Zstate
   @ 12,0 SAY '7. ZIP '
     @ 12,31 SAY Zzip
   @ 13,0 SAY '8. HOME PHONE '
     @ 13,31 SAY Zhphone
   @ 14,0 SAY '9. BUSINESS PHONE '
     @ 14,31 SAY Zbphone
   @ 15,0 SAY '10. SECURITY CLEARANCE '
     @ 15,31 SAY Zsecclr
   @ 16,0 SAY '11. GEO. ZONE '
     @ 16,31 SAY Zzone

   STORE ' ' to yesno
   @ 22,5 SAY 'Are these correct (Y/N)?'
   @ 22,30 GET yesno
    READ
   IF !(yesno) = 'Y'
      REPLACE name WITH Zname
      REPLACE rank WITH Zrank
      REPLACE ssn WITH Zssn
      REPLACE street WITH Zstreet
      REPLACE city WITH Zcity
      REPLACE state WITH Zstate
      REPLACE zip WITH Zzip
      REPLACE hphone WITH Zhphone
      REPLACE bphone WITH Zbphone
      REPLACE secclr WITH Zsecclr
      REPLACE zone WITH Zzone
   ELSE
      ERASE
      @ 5,12 SAY ' NO CHANGES MADE TO MEMBERs DATA'
      @ 7,12 SAY 'Hit SPACE BAR to EDIT more or to QUIT'
      SET CONSOLE OFF
        WAIT
      SET CONSOLE ON
   ENDIF
 LOOP
 ENDDO
 LOOP
 *EOF: ofixit1.prg


*************** EROSTER.PRG **************
* Enlisted Roster
* Author: M.W.Skahan
* Date: JAN 86
```

```
* Allows user to print out
*  an Alphabetized Roster of
* all Squadron enlisted personnel.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,15 SAY "HS-85 ENLISTED PERSONNEL ROSTER"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a ROSTER"
@  6,5 SAY "  for all enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,9 SAY "Any other key returns you to ENLISTED Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
@ 5,0 SAY ' Now putting ENLISTED personnel
 in alphabetical order...'
STORE driver+'enlisted' to drivefile
USE &drivefile
INDEX ON name to &LNAME
USE &drivefile INDEX &lname
@ 10,0 SAY ' Now transferring to printer.'
REPORT FORM eroster TO PRINT
RELEASE answer
RETURN
ENDDO
* EOF:  EROSTER.PRG
```

```
***************  ERECALL.PRG **************
* Enlisted Recall Bill
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  an Enlisted Recall Bill.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,15 SAY "HS-85 ENLISTED PERSONNEL RECALL BILL"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a RECALL BILL"
@  6,5 SAY "  for all enlisted personnel in the squadron."
@  8,5 SAY "      Press the  P  key to print."
@  9,5 SAY "       Any other key will return you to
 ENLISTED Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
@ 5,0 SAY ' Now putting ENLISTED personnel in
 alphabetical order...'
STORE driver+'enlisted' to drivefile
USE &drivefile
INDEX ON name to &LNAME
USE &drivefile INDEX &lname
@ 10,0 SAY ' Now transferring to printer.'
REPORT FORM erecall TO PRINT
RELEASE answer
```

```
RETURN
ENDDO
* EOF:  ERECALL.PRG

***************  ESENIOR.PRG **************
* Enlisted Seniority Listing
* Author:  M.W.Skahan
* Date:  JAN 86
* Allows user to print out
*   a Enlisted listing ordered Seniority (E-9 to E-1)
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@   1, 0 SAY "=========================================="
@   1,40 SAY "=========================================="
@   2, 0 SAY "||"
@   2,18 SAY "HS-85 ENLISTED SENIORITY LIST"
@   2,78 SAY "||"
@   3, 0 SAY "=========================================="
@   3,40 SAY "=========================================="
@   4, 0 SAY "||"
@   4,78 SAY "||"
@   5, 0 SAY "||"
@   5,78 SAY "||"
@   6, 0 SAY "||"
@   6,78 SAY "||"
@   7, 0 SAY "||"
@   7,78 SAY "||"
@   8, 0 SAY "||"
@   8,78 SAY "||"
@   9, 0 SAY "||"
@   9,78 SAY "||"
@  10, 0 SAY "||"
@  10,78 SAY "||"
@  11, 0 SAY "||"
@  11,78 SAY "||"
@  11, 0 SAY "=========================================="
@  11,40 SAY "=========================================="
@   5,5 SAY " This will print a Seniority
 listing (E-9 to E-1)"
@   6,5 SAY "  for all enlisted personnel in the squadron."
@   8,5 SAY "      Press the  P  key to print."
@   9,5 SAY "       Any other key will return you to
 ENLISTED Menu."
STORE ' ' to answer
@   8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
@ 5,0 SAY ' Now ordering personnel by RATE and Date
 of Rate...'
STORE driver+'enlisted' to drivefile
USE &drivefile
```

```
INDEX ON pgc+dor to &ZPGC
USE &drivefile INDEX &ZPGC
@ 10,0 SAY '  Now transferring to printer.'
REPORT FORM esenior TO PRINT
RELEASE answer
RETURN
ENDDO
* EOF: ESENIOR.PRG

***************  MANPOWER.PRG **************
* Enlisted Manpower
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  a Manpower listing ordered by Billet Sequence Code.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,18 SAY "HS-85 SQUADRON MANPOWER"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print the MANPOWER listing "
@  6,5 SAY "  for all enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you to
 ENLISTED Menu."
STORE "  " to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
```

```
@ 5,0 SAY ' Now putting ENLISTED personnel Billet
  Sequence order...'
STORE driver+'enlisted' to drivefile
USE &drivefile
INDEX ON bsc+rbsc to &ZBSC
USE &drivefile INDEX &ZBSC
@ 10,0 SAY ' Now transferring to printer.'
REPORT FORM manpower TO PRINT
RELEASE answer
RETURN
ENDDO
* EOF: MANPOWER.PRG

***************  ESECURE.PRG **************
* MENU for ENLISTED SECURITY PROGRAMS
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*   to process enlisted security data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,12 SAY "HS-85 ENLISTED SECURITY INFORMATION "
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "=========================================="
@ 13,40 SAY "=========================================="
@  5,20 SAY " 0. exit to ENLISTED MENU"
@  6,20 SAY " 1. Squadron Roster of all"
@  7,20 SAY " 2. Top Secret Clearances"
@  8,20 SAY " 3. Secret Clearances"
@  9,20 SAY " 4. Confidential Clearances"
@ 10,20 SAY " 5. Personnel without any clearance"
STORE  6 TO selectnum

DO WHILE selectnum < 0 .OR. selectnum >  5
   STORE " " TO select
   @ 12,33 SAY " select (0-5) : "
   @ 12,48 GET select PICTURE "#"
   READ
   STORE VAL(select) TO selectnum
ENDDO

DO CASE
```

```
    CASE selectnum= 0
       erase
       RETURN
    CASE selectnum= 1
    *  DO the squadron roster
       DO ESECALL
    CASE selectnum= 2
    *  DO enlisted with top secret clearances
       DO ESECTOP
    CASE selectnum= 3
    *  DO enlisted with secret clearances
       DO ESECSEC
    CASE selectnum= 4
    *  DO enlisted with confidential clearances
       DO ESECCONF
    CASE selectnum= 5
    *  DO enlisted with no clearance
       DO ESECNONE
    OTHERWISE
       ERASE
       @ 7,10 SAY CHR(7) + "INVALID ANSWER, TRY 0 - 5"
ENDCASE
?
?
STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: ESECURE.PRG

***************  ESECALL.PRG **************
* Enlisted Roster of all clearances
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  an Alphabetized Roster of all security clearances held.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@  2, 0 SAY "||"
@  2,15 SAY "HS-85 ENLISTED PERSONNEL CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "========================================="
@  3,40 SAY "========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "========================================="
@ 11,40 SAY "========================================="
@  5,5 SAY " This will print a listing of all clearances"
@  6,5 SAY "  held by enlisted personnel in the squadron."
```

```
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you to
 CLEARANCE Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF
ERASE
@ 5,0 SAY ' Now putting ENLISTED personnel in
 alphabetical order...'
STORE driver+'enlisted' to drivefile
USE &drivefile
INDEX ON name to &LNAME
USE &drivefile INDEX &lname
@ 10,0 SAY ' Now transferring clearances to printer.'
REPORT FORM esecall TO PRINT
RELEASE answer
RETURN
ENDDO
* EOF: ESECALL.PRG

***************  ESECTOP.PRG **************
* Enlisted Top Secret Listing
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  a Enlisted listing with TS clearance
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@  2, 0 SAY "||"
@  2,18 SAY "HS-85 ENLISTED TOP SECRET CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "========================================="
@  3,40 SAY "========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
```

```
@ 11, 0 SAY "========================================="
@ 11,40 SAY "========================================="
@  5,5 SAY " This will print a tcp secret clearance
 listing"
@  6,5 SAY "  for enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you to
 CLEARANCE Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
? '                                                  ',DATE()
? ' '              HS-85 TOP SECRET CLEARANCES'
? ' '
? 'Rate          Name                              SSN
     Clearance'
? ' '
SET PRINT OFF


ERASE
@ 5,0 SAY ' Now getting all personnel with top
 secret clearances...'

  STORE '                          ' to Zname
  STORE '       ' to Zrate
  STORE '                 to Zssn,
  STORE '                          to Zsecclr
   STORE driver+'enlisted' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   GOTO TOP
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rate TO Zrate
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'TOP SECRET'
      SET PRINT ON
      ? Zrate,    ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
   ENDIF
   SKIP
  ENDDO
SET PRINT ON
   EJECT
SET PRINT OFF
    *to clear buffer
```

```
RELEASE answer,Zname,Zrate,Zssn,Zsecclr
RETURN
ENDDO
* EOF: ESECTOP.PRG

***************  ESECSEC.PRG **************
* Enlisted Secret Listing
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*   a Enlisted listing with Secret clearances
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@  2, 0 SAY "||"
@  2,18 SAY "HS-85 ENLISTED SECRET CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "========================================="
@  3,40 SAY "========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "========================================="
@ 11,40 SAY "========================================="
@  5,5 SAY " This will print a secret clearance listing"
@  6,5 SAY "  for enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "       Any other key will return you to
 CLEARANCE Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
?                                                    ',DATE( )
```

```
? '                HS-85 SECRET CLEARANCES'
? ' '
? 'Rate          Name                                SSN
      Clearance'
? ' '
SET PRINT OFF


ERASE
@ 5,0 SAY ' Now getting all personnel with secret
 clearances...'
   STORE '                              ' to Zname
   STORE '       ' to Zrate
   STORE '                 to Zssn
   STORE '                              ' to Zsecclr
    STORE driver+'enlisted' TO drivefile
    USE &drivefile
    index on name to &lname
    use &drivefile index &lname

    GOTO TOP
   DO WHILE .not. EOF
    STORE name TO Zname
    STORE rate TO Zrate
    STORE ssn TO Zssn
    STORE secclr TO Zsecclr
    STORE TRIM(!(Zsecclr)) TO Zsecclr
    IF Zsecclr = 'SECRET'
       SET PRINT ON
       ? Zrate,'  ',Zname,' ',Zssn,'    ',Zsecclr
       SET PRINT OFF
    ENDIF
    SKIP
   ENDDO
SET PRINT ON
    EJECT
SET PRINT OFF
     *to clear buffer
RELEASE answer,Zname,Zrate,Zssn,Zsecclr
RETURN
ENDDO
* EOF: ESECSEG.PRG

*************** ESECCONF.PRG **************
* Enlisted Confidential Listing
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*  a Enlisted listing with Confidential clearances
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@  2, 0 SAY "||"
@  2,18 SAY "HS-85 ENLISTED CONFIDENTIAL CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "========================================="
@  3,40 SAY "========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
```

```
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "========================================="
@ 11,40 SAY "========================================="
@  5,5 SAY " This will print a confidential
 clearance listing"
@  6,5 SAY "  for enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you to
 CLEARANCE Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
? '                                                ',DATE()
? '              HS-85 CONFIDENTIAL CLEARANCES'
? ' '
? 'Rate          Name                              SSN
      Clearance'
? ' '
SET PRINT OFF
ERASE
@ 5,0 SAY ' Now getting all personnel with confidential'

  STORE '                          ' to Zname
  STORE '       ' to Zrate
  STORE '                  to Zssn
  STORE '                          to Zsecclr
   STORE driver+'enlisted' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   GOTO TOP
  DO WHILE .not. EOF
   STORE name TO Zname
   STORO Zrate
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'CONFIDENTIAL'
      SET PRINT ON
      ? Zrate,'   ',Zname,' ',Zssn,'   ',Zsecclr
      SET PRINT OFF
```

```
   ENDIF
   SKIP
  ENDDO
SET PRINT ON
   EJECT
SET PRINT OFF
    *to clear buffer
RELEASE answer,Zname,Zrate,Zssn,Zsecclr
RETURN
ENDDO
* EOF: ESECCONF.PRG

***************  ESECNONE.PRG **************
* Enlisted With No Security clearance
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to print out
*   a Enlisted listing with NO clearances
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
DO WHILE T
ERASE
@  1, 0 SAY "=========================================="
@  1,40 SAY "=========================================="
@  2, 0 SAY "||"
@  2,18 SAY "HS-85 ENLISTED WITH NO CLEARANCES"
@  2,78 SAY "||"
@  3, 0 SAY "=========================================="
@  3,40 SAY "=========================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 11, 0 SAY "=========================================="
@ 11,40 SAY "=========================================="
@  5,5 SAY " This will print a no-clearance listing"
@  6,5 SAY "  for enlisted personnel in the squadron."
@  8,5 SAY "     Press the  P  key to print."
@  9,5 SAY "      Any other key will return you to
 CLEARANCE Menu."
STORE ' ' to answer
@  8,7 get answer
 READ
if !(answer) <> "P"
   RETURN
ENDIF
STORE "P" to answer
ERASE
@ 10,0 SAY '    Make sure your PRINTERs "ON LINE" light
 is lit, then'
@ 12,0 SAY '     HIT ENTER'
@ 14,0 SAY '    Or to QUIT NOW, press any other key.'
@ 16,10 GET answer
READ
IF !(answer) <> "P"
```

```
   RETURN
ENDIF

ERASE
SET PRINT ON
EJECT
? '                                                    ',DATE()
? '          HS-85 PERSONNEL WITH NO SECURITY CLEARANCE'
? ' '
? 'Rate          Name                              SSN
   Clearance'
? ' '
SET PRINT OFF
ERASE
@ 5,0 SAY ' Now getting all personnel with
 no clearances...'

  STORE '                          ' to Zname
  STORE '        ' to Zrate
  STORE '                  to Zssn
  STORE '                          to Zsecclr
   STORE driver+'enlisted' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   GOTO TOP
  DO WHILE .not. EOF
   STORE name TO Zname
   STORE rate TO Zrate
   STORE ssn TO Zssn
   STORE secclr TO Zsecclr
   STORE TRIM(!(Zsecclr)) TO Zsecclr
   IF Zsecclr = 'NONE'
      SET PRINT ON
      ? Zrate,'  ',Zname,' ',Zssn,'    ',Zsecclr
      SET PRINT OFF
   ENDIF
   SKIP
  ENDDO
SET PRINT ON
   EJECT
SET PRINT OFF
    *to clear buffer
RELEASE answer,Zname,Zrate,Zssn,Zsecclr
RETURN
ENDDO
* EOF: ESECNONE.PRG

***************  UPDATE.PRG *************
* UPDATE MENU for ENLISTED PROGRAMS
* Author: M.W.Skahan
* Date: JAN 86
* Provides user with choice of options
*  to update enlisted data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF

ERASE
* draw the screen with choices
DO WHILE T
ERASE
@  1, 0 SAY "========================================="
@  1,40 SAY "========================================="
@@ 2, 0 SAY "||"
@@ 2,12 SAY "HS-85 ENLISTED DATA BASE UPDATE SELECTIONS"
@@ 2,78 SAY "||"
@  3, 0 SAY "========================================="
```

```
@  3,40 SAY "======================================="
@  4, 0 SAY "||"
@  4,78 SAY "||"
@  5, 0 SAY "||"
@  5,78 SAY "||"
@  6, 0 SAY "||"
@  6,78 SAY "||"
@  7, 0 SAY "||"
@  7,78 SAY "||"
@  8, 0 SAY "||"
@  8,78 SAY "||"
@  9, 0 SAY "||"
@  9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "======================================="
@ 13,40 SAY "======================================="
@  5,20 SAY " 0. exit to ENLISTED Programs "
@  6,20 SAY " 1. ADD a new squadron member"
@  7,20 SAY " 2. DELETE a squadron member"
@  8,20 SAY " 3. CHANGE an existing member's data"
STORE 4 TO selectnum

* get & process the selection
DO WHILE selectnum < 0 .OR. selectnum >  3
   STORE " " TO select
   @ 12,33 SAY " select (0-3) : : "
   @ 12,48 GET select PICTURE "#"
   READ
   STORE VAL(select) TO selectnum
ENDDO

DO CASE
   CASE selectnum= 0
     *GO BACK TO ENLISTED MENU
      ERASE
      CLEAR
      RETURN
   CASE selectnum= 1
   *  Do the addition of a squadron member
      DO EADD
   CASE selectnum= 2
   *  Do the deletion of a member
      DO EDELETE
   CASE selectnum= 3
   *  Do a change to the data
      DO EFIXIT2
   OTHERWISE
      erase
      @ 7,10 SAY CHR(7) + "INVALID ANSWER, TRY 0 - 3"
ENDCASE

STORE " " TO select
@ 23,0 SAY "Strike any key to continue... " GET select
READ
ENDDO T
* EOF: UPDATE.PRG

***************  EADD.PRG **************
* ADD ENLISTED DATA
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to add
*   enlisted data.
*****************************************
SET TALK OFF
```

```
SET INTENSITY OFF
SET BELL OFF
store Y to addmore
ERASE
@ 1,26 say 'H S  -  85'
@ 2,22 say 'A D D   D A T A   TO  '
@ 3,15 say 'E N L I S T E D   P E R S O N N E L'
STORE 'N' TO YESNO
@ 6,8 SAY "Press the Y key to see an explanation
 of the data fields."
@ 7,9 SAY "Otherwise just hit the space bar."
@ 8,8 GET YESNO
READ
IF !(YESNO) = 'Y'
   DO DATADIC2
ENDIF

DO WHILE addmore
ERASE
STORE driver+'edummy' TO drivefile
USE &drivefile
DELETE ALL
PACK
APPEND BLANK

@ 1,0 SAY '        HS-85'
@ 2,0 SAY 'ADD NEW PERSONNEL DATA'
@ 4,0 SAY '1. BSC ' GET bsc PICTURE '!!!!!!!'
@ 5,0 SAY '2. RBSC ' GET rbsc PICTURE '!!!!!!!'
@ 6,0 SAY '3. LAST NAME, FIRST NAME, MI ' GET name;
 PICTURE '!!!!!!!!!!!!!!!!!!!!'

@ 7,0 SAY '4. ASSIGNED RATE ' GET arate PICTURE '!!!!!!!'
@ 7,30 SAY '5. ACTUAL RATE ' GET rate PICTURE '!!!!!!!'
@ 8,0 SAY '6. SSN ' GET ssn PICTURE '!!!!!!!!!!!'
@ 9,0 SAY '7. PNEC ' GET pnec PICTURE '!!!!!!!'
@ 9,30 SAY '8. SNEC ' GET snec PICTURE '!!!!!!!'
@ 10,0 SAY '9. EAOS ' GET eaos PICTURE '!!!!!!'
@ 11,0 SAY '10. PRD ' GET prd PICTURE '!!!!!!'
@ 11,30 SAY '11. DATE of RATE ' GET dor PICTURE '!!!!!!'
@ 12,0 SAY '12. STATUS ' GET status PICTURE '!!!!!!'
@ 13,0 SAY '13. DEPARTMENT ' GET dept
 PICTURE '!!!!!!!!!!!!'
@ 14,0 SAY '14. PAY GRADE CODE ' GET pgc PICTURE '!!!!'
@ 15,0 SAY '15. STREET ' GET street
 PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
@ 16,0 SAY '16. CITY ' GET city PICTURE '!!!!!!!!!!!!!!!!'
@ 17,0 SAY '17. STATE ' GET state PICTURE '!!!!!!'
@ 17,30 SAY '18. ZIP ' GET zip PICTURE '!!!!!!!!!!'
@ 18,0 SAY '19. HOME PHONE ' GET hphone
 PICTURE '!!!!!!!!!!!!!!'
@ 18,30 SAY '20. BUSINESS PHONE ' GET bphone
 PICTURE '!!!!!!!!!!!!!!'
@ 19,0 SAY '21. SECURITY CLEARANCE ';
   GET seccl r PICTURE '!!!!!!!!!!!!!!!!!!!!'
@ 20,0 SAY '22. GEO. ZONE ' GET zone PICTURE '!!!!'

STORE ' ' TO loopit
DO WHILE loopit=' '
STORE ' ' TO decide
@ 23,25 SAY 'MAIN MENU = M, CANCEL = C, PROCESS = P' GET
 decide PICTURE '!'
READ
IF !(decide)='M' .OR. !(decide)='C' .OR. !(decide)='P'
   STORE '!' TO loopit
ENDIF
ENDDO

IF !(decide) = 'P'
  STORE driver+'enlisted' TO drivefile
```

```
   USE &drivefile
   STORE driver+'edummy' TO addum
   APPEND FROM &addum
   @ 22,45 SAY 'FUNCTION COMPLETED'
   STORE ' ' TO loopit

   DO WHILE loopit=' '
    STORE ' ' TO decide
    @ 23,25 SAY 'MAIN MENU = M, CANCEL = C, PROCESS = P';
      GET decide PICTURE '!'
    READ
    IF !(decide)='M' .OR. !(decide)='C' .OR. !(decide)='P'
      STORE 'X' TO loopit
    ENDIF
   ENDDO
    IF !(decide) = 'P'
          STORE Y TO addmore
    ELSE
          STORE N TO addmore
    ENDIF decide
ELSE
     STORE N TO addmore
ENDIF decide
ENDDO addmore

RELEASE addum,decide,loopit,addmore
USE
STORE driver+'enlisted' TO drivefile
RETURN
* EOF: EADD.PRG

*************** EDELETE.PRG **************
* DELETE ENLISTED RECORDS
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to delete
*  enlisted records from data base..
*****************************************
SET TALK OFF
SET BELL OFF
ERASE
STORE Y TO dropum
DO WHILE dropum
   ERASE
   STORE driver+'enlisted' TO drivefile
   USE &drivefile
   INDEX on name to &lname
    USE &drivefile index &lname

   @ 1,0 SAY '         HS-85'
   @ 2,0 SAY 'DELETE SQUADRON MEMBER'

   STORE '                          ' TO member
   STORE '        ' TO Zbsc
   STORE '        ' TO Zrbsc
   STORE '                          ' TO Zname
   STORE '        ' TO Zarate
   STORE '        ' TO Zrate
   STOR         ' TO Zssn
   STORE '        ' TO Zpnec
   STORE '        ' TO Zsnec
   STORE '        ' TO Zeaos
   STORE '        ' TO Zprd
   STORE '        ' TO Zdor
   STORE '        ' TO Zstatus
   STORE '             ' TO Zdept
   STORE '    ' TO Zpgc
   STORE '                          ' TO Zstreet
   STORE '                    ' TO Zcity
   STORE '     ' TO Zstate
```

```
    STORE '           ' TO Zzip
    STORE '              ' TO Zhphone
    STORE '              ' TO Zbphone
    STORE '                        ' TO Zsecclr
    STORE '     ' TO Zzone

      @ 6,22 say "To EXIT, push ENTER only."
      @ 4,0 SAY "Who do you want to DELETE ?"
        GET member PICTURE '!!!!!!!!!!!!!!!!!!!!'
      READ
      IF member = ' '
        RETURN
      ENDIF
    STORE TRIM(!(member)) to name
    FIND &NAME

    STORE # to recnum
erase
  IF recnum = 0
    @ 10,20 SAY 'That spelling NOT found,'
    @ 12,15 SAY 'try again.  A short version like
      SMI  for SMITH'
    @ 14,15 SAY 'is OK to use.'
    @ 18,5 SAY 'HIT ANY KEY to continue.'
    set console OFF
      WAIT
    set console ON
  ELSE

    STORE bsc TO Zbsc
    STORE rbsc TO Zrbsc
    STORE name TO Zname
    STORE arate TO Zarate
    STORE rate TO Zrate
    STORE ssn TO Zssn
    STORE pnec TO Zpnec
    STORE snec TO Zsnec
    STORE eaos TO Zeaos
    STORE prd TO Zprd
    STORE dor TO Zdor
    STORE status TO Zstatus
    STORE dept TO Zdept
    STORE pgc TO Zpgc
    STORE street TO Zstreet
    STORE city TO Zcity
    STORE state TO Zstate
    STORE zip TO Zzip
    STORE hphone TO Zhphone
    STORE bphone TO Zbphone
    STORE secclr TO Zsecclr
    STORE zone TO Zzone
     @ 1,0 SAY '                HS-85   DELETE PERSONNEL DATA'
     @ 3,0 SAY '     This will DELETE this person
      from the records.'

    @ 4,0 SAY '1. BSC '
      @ 4,8 SAY Zbsc
    @ 5,0 SAY '2. RBSC '
      @ 5,9 SAY Zrbsc
    @ 6,0 SAY '3. NAME '
      @ 6,9 SAY Zname
    @ 7,0 SAY '4. ASSIGNED RATE '
      @ 7,17 SAY Zarate
    @ 7,30 SAY '5. ACTUAL RATE '
      @ 7,46 SAY Zrate
    @ 8,0 SAY '6. SSN '
      @ 8,8 SAY Zssn
    @ 9,0 SAY '7. PNEC '
      @ 9,9 SAY Zpnec
    @ 9,30 SAY '8. SNEC '
```

```
    @ 9,39 SAY Zsnec
  @ 10,0 SAY '9. EAOS '
    @ 10,9 SAY Zeaos
  @ 11,0 SAY '10. PRD '
    @ 11,8 SAY Zprd
  @ 11,30 SAY '11. DATE of RATE '
    @ 11,19 SAY Zdor
  @ 12,0 SAY '12. STATUS '
    @ 12,12 SAY Zstatus
  @ 13,0 SAY '13. DEPARTMENT '
    @ 13,16 SAY Zdept
  @ 14,0 SAY '14. PAY GRADE CODE '
    @ 14,22 SAY Zpgc
  @ 15,0 SAY '15. STREET '
    @ 15,13 SAY Zstreet
  @ 16,0 SAY '16. CITY '
    @ 16,11 SAY Zcity
  @ 17,0 SAY '17. STATE '
    @ 17,12 SAY Zstate
  @ 17,30 SAY '18. ZIP '
    @ 17,39 SAY Zzip
  @ 18,0 SAY '19. HOME PHONE '
    @ 18,18 SAY Zhphone
  @ 18,30 SAY '20. BUSINESS PHONE '
    @ 18,50 SAY Zbphone
  @ 19,0 SAY '21. SECURITY CLEARANCE '
    @ 19,25 SAY Zsecclr
  @ 20,0 SAY '22. GEO. ZONE '
    @ 20,17 SAY Zzone
**************************
@ 22,33 SAY 'To DELETE this person type "D"'
STORE ' ' TO loopit
DO WHILE loopit=' '
    STORE ' ' TO decide
    @ 23,20 SAY 'MAIN MENU = M, CANCEL = C, DELETE = D'
    @ 23,60 GET decide
    READ
    IF !(decide)='M' .OR. !(decide)='C'
     .OR. !(decide)='D'
      STORE 'X' TO loopit
    ENDIF
ENDDO

IF !(decide) = 'D'
    DELETE
    PACK
    STORE ' ' TO decide
    @ 22,12 SAY 'FUNCTION COMPLETED:'
    STORE ' ' TO loopit
    DO WHILE loopit=' '
       STORE ' ' TO decide
        @ 23,20 SAY 'UP DATE MENU = M,
         CANCEL = C, DO MORE = D'
        @ 23,63 GET decide
        READ
       IF !(decide)='M' .OR. !(decide)='C'
        .OR. !(decide)='D'
          STORE 'X' TO loopit
       ENDIF
    ENDDO
    IF !(decide) = 'D'
              STORE Y TO dropum
    ELSE
              STORE N TO dropum
    ENDIF decide
 ELSE
    STORE N TO dropum
 ENDIF decide
 USE
ENDDO dropum
```

```
USE
RELEASE dropum,member,loopit,decide
RETURN
* EOF: EDELETE.PRG

***************  EFIXIT2.PRG **************      fixit2
* FIX ENLISTED DATA
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to change values in
*   enlisted data.
****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
STORE 0 TO recnum
DO WHILE t
   ERASE
   @ 1,0 SAY 'Getting ENLISTED data....'
   STORE driver+'enlisted' TO drivefile
   USE &drivefile
   index on name to &lname
   use &drivefile index &lname

   @ 1,0 SAY '       HS-85                    '
   @ 2,0 SAY 'CHANGE PERSONNEL DATA'
   STORE '                              ' TO member
   STORE '         ' TO Zbsc
   STORE '         ' TO Zrbsc
   STORE '                              ' TO Zname
   STORE '         ' TO Zarate
   STORE '         ' TO Zrate
   STORE '                   ' TO Zssn
   STORE '         ' TO Zpnec
   STORE '         ' TO Zsnec
   STORE '         ' TO Zeaos
   STORE '         ' TO Zprd
   STORE '         ' TO Zdor
   STORE '         ' TO Zstatus
   STORE '                ' TO Zdept
   STORE '     ' TO Zpgc
   STORE '                              ' TO Zstreet
   STORE '                         ' TO Zcity
   STORE '      ' TO Zstate
   STORE '           ' TO Zzip
   STORE '                ' TO Zhphone
   STORE '                ' TO Zbphone
   STORE '                              ' TO Zsecclr
   STORE '     ' TO Zzone

     @ 6,22 say "To EXIT, push ENTER only."
     @ 4,0 SAY "Who's data needs update?";
       GET member PICTURE '!!!!!!!!!!!!!!!!!!!!!!!'
     READ
     IF member = ' '
        RETURN
     ENDIF
   STORE TRIM(!(member)) to name
   FIND &NAME

   STORE # to recnum
erase
  IF recnum = 0
    @ 10,30 SAY 'That spelling NOT found,'
    @ 12,26 SAY 'try again.  Push any key to continue.'
    set console OFF
      WAIT
    set console ON
  ELSE
```

```
      STORE bsc TO Zbsc
      STORE rbsc TO Zrbsc
      STORE name TO Zname
      STORE arate TO Zarate
      STORE rate TO Zrate
      STORE ssn TO Zssn
      STORE pnec TO Zpnec
      STORE snec TO Zsnec
      STORE eaos TO Zeaos
      STORE prd TO Zprd
      STORE dor TO Zdor
      STORE status TO Zstatus
      STORE dept TO Zdept
      STORE pgc TO Zpgc
      STORE street TO Zstreet
      STORE city TO Zcity
      STORE state TO Zstate
      STORE zip TO Zzip
      STORE hphone TO Zhphone
      STORE bphone TO Zbphone
      STORE secclr TO Zsecclr
      STORE zone TO Zzone
       @ 1,0 SAY '             HS-85   CHANGE PERSONNEL DATA'
       @ 2,0 SAY '              If entry is good, hit ENTER.'
       @ 3,0 SAY '    To make CHANGE, Type over old
        entries and ENTER.'
  @ 4,0 SAY '1. BSC ' SAY Zbsc
    @ 4,7 get Zbsc PICTURE '!!!!!!!'
  @ 5,0 SAY '2. RBSC ' SAY Zrbsc
    @ 5,8 get Zrbsc PICTURE '!!!!!!!'
  @ 6,0 SAY '3. LAST NAME, FIRST NAME, MI ' SAY Zname
    @ 6,29 get Zname PICTURE '!!!!!!!!!!!!!!!!!!!!!'

  @ 7,0 SAY '4. ASSIGNED RATE ' SAY Zarate
    @ 7,17 get Zarate PICTURE '!!!!!!!'

  @ 7,30 SAY '5. ACTUAL RATE ' SAY Zrate
    @ 7,45 get Zrate PICTURE '!!!!!!!'

  @ 8,0 SAY '6. SSN ' SAY Zssn
    @ 8,7 get Zssn PICTURE '!!!!!!!!!!!!'

  @ 9,0 SAY '7. PNEC ' SAY Zpnec
    @ 9,8 get Zpnec PICTURE '!!!!!!'

  @ 9,30 SAY '8. SNEC ' SAY Zsnec
    @ 9,38 get Zsnec PICTURE '!!!!!!'

  @ 10,0 SAY '9. EAOS ' SAY Zeaos
    @ 10,8 get Zeaos PICTURE '!!!!!!'

  @ 11,0 SAY '10. PRD ' SAY Zprd
    @ 11,8 get Zprd PICTURE '!!!!!!'

  @ 11,30 SAY '11. DATE of RATE ' SAY Zdor
    @ 11,47 get Zdor PICTURE '!!!!!!'

  @ 12,0 SAY '12. STATUS ' SAY Zstatus
    @ 12,11 get Zstatus PICTURE '!!!!!!'

  @ 13,0 SAY '13. DEPARTMENT ' SAY Zdept
    @ 13,15 get Zdept PICTURE '!!!!!!!!!!!!'

  @ 14,0 SAY '14. PAY GRADE CODE ' SAY Zpgc
    @ 14,19 get Zpgc PICTURE '!!!!'

  @ 15,0 SAY '15. STREET ' SAY Zstreet
    @ 15,11 get Zstreet PICTURE '!!!!!!!!!!!!!!!!!!!!'
```

```
@ 16,0 SAY '16. CITY ' SAY Zcity
  @ 16,9 get Zcity PICTURE '!!!!!!!!!!!!!!!!'

@ 17,0 SAY '17. STATE ' SAY Zstate
  @ 17,10 get Zstate PICTURE '!!!!!'

@ 17,30 SAY '18. ZIP ' SAY Zzip
  @ 17,38 get Zzip PICTURE '!!!!!!!!!!'

@ 18,0 SAY '19. HOME PHONE ' SAY Zhphone
  @ 18,15 get Zhphone PICTURE '!!!!!!!!!!!!!!'

@ 18,30 SAY '20. BUSINESS PHONE ' SAY Zbphone
  @ 18,49 get Zbphone PICTURE '!!!!!!!!!!!!!!!'

@ 19,0 SAY '21. SECURITY CLEARANCE ' SAY Zsecclr
  @ 19,23 get Zsecclr PICTURE '!!!!!!!!!!!!!!!!!!!!!!'

@ 20,0 SAY '22. GEO. ZONE ' SAY Zzone
  @ 20,14 get Zzone PICTURE '!!!!'
READ
IF Zname = '                              '
   RETURN
ENDIF
ERASE
   @ 1,0 SAY '  Your new changes look like this:'
   @ 4,0 SAY '1. BSC '
     @ 4,8 SAY Zbsc
   @ 5,0 SAY '2. RBSC '
     @ 5,9 SAY Zrbsc
   @ 6,0 SAY '3. LAST NAME, FIRST NAME, MI '
     @ 6,31 SAY Zname
   @ 7,0 SAY '4. ASSIGNED RATE '
     @ 7,17 SAY Zarate
   @ 7,30 SAY '5. ACTUAL RATE '
     @ 7,46 SAY Zrate
   @ 8,0 SAY '6. SSN '
     @ 8,8 SAY Zssn
   @ 9,0 SAY '7. PNEC '
     @ 9,9 SAY Zpnec
   @ 9,30 SAY '8. SNEC '
     @ 9,39 SAY Zsnec
   @ 10,0 SAY '9. EAOS '
     @ 10,9 SAY Zeaos
   @ 11,0 SAY '10. PRD '
     @ 11,8 SAY Zprd
   @ 11,30 SAY '11. DATE of RATE '
     @ 11,19 SAY Zdor
   @ 12,0 SAY '12. STATUS '
     @ 12,12 SAY Zstatus
   @ 13,0 SAY '13. DEPARTMENT '
     @ 13,16 SAY Zdept
   @ 14,0 SAY '14. PAY GRADE CODE '
     @ 14,22 SAY Zpgc
   @ 15,0 SAY '15. STREET '
     @ 15,13 SAY Zstreet
   @ 16,0 SAY '16. CITY '
     @ 16,11 SAY Zcity
   @ 17,0 SAY '17. STATE '
     @ 17,12 SAY Zstate
   @ 17,30 SAY '18. ZIP '
     @ 17,39 SAY Zzip
   @ 18,0 SAY '19. HOME PHONE '
     @ 18,18 SAY Zhphone
   @ 18,30 SAY '20. BUSINESS PHONE '
     @ 18,50 SAY Zbphone
   @ 19,0 SAY '21. SECURITY CLEARANCE '
     @ 19,25 SAY Zsecclr
   @ 20,0 SAY '22. GEO. ZONE '
     @ 20,17 SAY Zzone
```

```
    STORE ' ' to yesno
    @ 22,5 SAY 'Are these correct (Y/N)?'
    @ 22,30 GET yesno
     READ
    IF !(yesno) = 'Y'
       REPLACE bsc WITH Zbsc
       REPLACE rbsc WITH Zrbsc
       REPLACE name WITH Zname
       REPLACE arate WITH Zarate
       REPLACE rate WITH Zrate
       REPLACE ssn WITH Zssn
       REPLACE pnec WITH Zpnec
       REPLACE snec WITH Zsnec
       REPLACE eaos WITH Zeaos
       REPLACE prd WITH Zprd
       REPLACE dor WITH Zdor
       REPLACE status WITH Zstatus
       REPLACE dept WITH Zdept
       REPLACE pgc WITH Zpgc
       REPLACE street WITH Zstreet
       REPLACE city WITH Zcity
       REPLACE state WITH Zstate
       REPLACE zip WITH Zzip
       REPLACE hphone WITH Zhphone
       REPLACE bphone WITH Zbphone
       REPLACE secclr WITH Zsecclr
       REPLACE zone WITH Zzone
    ELSE
       ERASE
       @ 5,9 SAY '    NO CHANGES MADE TO MEMBERs DATA'
       @ 7,12 SAY '  Hit SPACE BAR to EDIT more or to QUIT'
       SET CONSOLE OFF
          WAIT
       SET CONSOLE ON
    ENDIF
  LOOP
  ENDDO
  LOOP
  *EOF: efixit2.prg

*************** DATADIC1.PRG **********
* ADD RECORD HELP
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to see data structure definitions
*    for adding to the officer data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
@ 1,26 say 'H S  -  85'
@ 2,22 say 'A D D   R E C O R D  H E L P'

? " "
? "NAME: Last, first, mi              22 letters maximum"
? "RANK: Rank                          4 letter abrev. max"
? "SSN:  Social Security number        9 digits "
? "                                      form:  XXX-XX-XXXX"
? "STREET: Street number & name       22 characters maximum"
? "CITY: City                         17 characters maximum"
? "STATE: State                        2 letter abreviation"
? "ZIP: 6 or 9 number zip              6 or 9 digits "
? "HPHONE: Home telephone             14 digits maximum"
? "                                      'none' permitted"
? "BPHONE: Business telehone          14 digits maximum"
? "                                      'none' permitted"
? "SECCLR: Security clearance         12 characters max: "
? "           Top Secret, Secret, Confidential, None "
```

```
? "ZONE: Geograph. home location     4 digit number max."

@ 23,25 SAY 'Hit SPACE BAR to continue.'
SET CONSOLE OFF
   WAIT
SET CONSOLE ON
RETURN
* EOF: DATADIC1.PRG

*************  DATADIC2.PRG **********
* ADD RECORD HELP
* Author: M.W.Skahan
* Date: JAN 86
* Allows user to see data structure definitions
*     for adding to the enlisted data.
*****************************************
SET TALK OFF
SET INTENSITY OFF
SET BELL OFF
ERASE
@ 1,26 say 'H S  -  85'
@ 2,22 say 'A D D   R E C O R D  H E L P'

? " "
? "BSC: Billet sequence code          7 digits"
? "RBSC: Reserve billet seq. code     7 digits"
? "NAME: Last, first, mi             22 letters maximum"
? "ARATE: Assigned rate               7 characters "
? "RATE:  Actual rate                 7 characters "
? "SSN:  Social Security number       9 digits "
? "                                     form:  XXX-XX-XXXX"
? "PNEC: Primary NEC                  6 digits "
? "SNEC: Secondary NEC                6 digits"
? "EAOS: End of current enlistment    6 digit date:  103185"
? "                                     for OCT 31, 1985"
? "PRD: Projected rotation date       6 digit date"
? "DOR: Date of rate                  6 digit data"
? "PGC: Pay grade code                2 digit num. used"
? "                          for ordering for E-9 to E-1: "
? "     09 = E-9, 18 = E-8, 27 = E-7, 36 = E-6, 45 = E-5,"
? "        54 = E-4, 63 = E-3, 72 = E-2, 81 = E-1"
?  "  "
@ 23,25 SAY 'Hit SPACE BAR to continue.'
SET CONSOLE OFF
   WAIT
SET CONSOLE ON
ERASE
? " "
? "STATUS: Reserve status             6 characters:"
? "                                     TAR, NONTAR, SELRES"
? "DEPT: Department assigned to      12 characters maximum"
? "STREET: Street number & name      22 characters maximum"
? "CITY: City                        17 characters maximum"
? "STATE: State                       2 letter abreviation"
? "ZIP: 6 or 9 number zip       6 or 9 digits "
? "HPHONE: Home telephone            14 digits maximum"
? "                                     'none' permitted"
? "BPHONE: Business telehone         14 digits maximum"
? "                                     'none' permitted"
? "SECCLR: Security clearance        12 characters max:"
? "          Top Secret, Secret, Confidential, None "
? "ZONE: Geograph. home location  4 digit number maximum"

@ 23,25 SAY 'Hit SPACE BAR to continue.'
SET CONSOLE OFF
   WAIT
SET CONSOLE ON
RETURN

* EOF: DATADIC2.PRG
```

```
****************** QUIT.PRG *****************
* PROGRAM TO QUIT THE DATA BASE
* Author: M.W.Skahan
* Date: JAN 86
* Farewell banner with instructions
*   on care of disks.
*********************************************
erase
@ 5,5 say "Leaving Personnel Management System"
@ 7,5 say "When the  A>  shows, remove the floppy
 disks.  Be careful not"
@ 8,5 say "to touch the brown part of the disk inside
 the cutout window."
@ 9,5 say "Return the disks to their jackets.  The
 information"
@ 10,5 say "on the disks is subject to the Privacy Act."
@ 11,5 say "Store the them in locked file cabinets."
@ 15,5 say "PRESS ANY KEY to QUIT."
set console off
wait
set console on
quit

* EOF: QUIT.PRG
* END OF PERSONNEL PROGRAM ***
```

2. DATA BASE ADMINISTRATOR INFORMATION

1. Module Descriptions

Note: Officer modules begin with "O" or end with "1".
Enlisted modules begin with "E" or end with "2".

1. GREET: Opening screens to identify program and its functions. Gets disk drive with data base. Provides Main Menu selection between Officer and Enlisted programs.

called by: AUTOEXEC.BAT
calls: MAIN1 (officers)
or MAIN2 (enlisted)
or QUIT (exit to DOS)
vars: driver, loopit, enl, offr, selectnum, select

2. MAIN1: Menu screen to select Officer programs or go to Enlisted menu through Greeting menu.

called by: GREET
calls: OSENIOR (list by ranks)
or OSECURE (security listing)
or ORECALL (officer recall)
or UPDATE1 (menu for data base updates)
or GREET (return)
vars: selectnum, select

3. MAIN2: Menu screen to select Enlisted programs or go to Officer menu through Greeting menu.

called by: GREET

calls: EROSTER (alphabetical listing of enlisted)
or ESENIOR (lists E-9 down to E-1)
or ESECURE (menu for each level of security clearance)
or ERECALL (recall bill)
or MANPOWER (lists squadron by billet sequence code)
or UPDATE2 (menu for data base updates)
or GREET (return)

vars: selectnum, select

4. OSENIOR: Provides a printout of officers listed by rank (CDR-CWO)

called by: MAIN1

calls: MAIN1 (return)

vars: answer, Zname, Zrank, Zssn, driver, drivefile, lname, cnt

5. OSECURE: Provides a printout of officer security clearances.

called by: MAIN1

calls: MAIN1 (return)

vars: answer, Zname, Zrank, Zseccir, driver, drivefile, lname, cnt

6. ORECALL: Prints out officer's names, addresses, and phone numbers.

called by: MAIN1

calls: MAIN1 (return)

vars: answer, driver, drivefile

7. UPDATE1: Menu for adding, deleting, changing data.

called by: MAIN1

calls: OADD (add officers to data base)
or ODELETE (drop officers from data base)
or OFIXIT1 (change contents of a record)

or QUIT (to DOS)

vars: selectnum, select

8. ODELETE: Allows deletion of an officer form the data base.

called by: UPDATE1

calls: UPDATE1 (return)

vars: member, dropum, driver, drivefile, lname recnum, decide, all Zofficer

9. OFIXIT1: Permits user to change individual fields of a selected officer's data.

called by: UPDATE1

calls: UPDATE1 (return)

vars: member, yesno, driver, drivefile, lname recnum, decide, all Zofficer

10. OADD: Permits user to add a new officer to the data base.

called by: UPDATE1

calls: UPDATE1 (return)

vars: addmore, yesno, driver, drivefile, loopit, decide, addum

11. EROSTER: Prints alphabetical listing of enlisted squadron members.

called by: MAIN2

calls: MAIN2 (return)

vars: answer, driver, drivefile, lname

12. ESENIOR: Prints sorted list of E-9 to E-1.

called by: MAIN2

calls: MAIN2 (return)

vars: answer, driver, drivefile, Zpgc

13. ESECURE: Menu to select which security clearances to print.

called by: MAIN2

calls: ESECALL (all clearances)

or ESECTOP (top secret)

or ESECSEC (secret)

or ESECCONF (confidential)
or ESECNONE (no clearance)
or ESECURE (return)

vars: selectnum, select

14. ERECALL: Prints out alphabetical recall list with names, addresses, phone numbers, and departments.

called by: MAIN2

calls: MAIN2 (return)

vars: answer, driver, drivefile, lname

15. MANPOWER: Prints enlisted by BSC and RBSC

called by: MAIN2

calls: MAIN2 (return)

vars: answer, driver, drivefile, Zbsc

16. UPDATE2: Menu for adding, deleting, changing data.

called by: MAIN2

calls: EADD (add enlisted to data base)
or EDELETE (drop enlisted from data base)
or EFIXIT2 (change contents of a record)
or QUIT (to DOS)

vars: selectnum, select

17. ESECALL: Prints listing of all enlisted and clearances held.

called by: ESECURE

calls: ESECURE (return)

vars: answer, driver, drivefile, lname

18. ESECTOP: Prints personnel holding top secret clearances.

called by: ESECURE

calls: ESECURE (return)

vars: answer, driver, drivefile, lname
Zname, Zrate, Zssn, Zsecclr

19. ESECSEC: Prints personnel holding secret clearances.

called by: ESECURE

calls: ESECURE (return)

vars: answer, driver, drivefile, lname
Zname, Zrate, Zssn, Zsecclr

20. ESECCONF: Prints personnel holding confidential clearances.
called by: ESECURE
calls: ESECURE (return)
vars: answer, driver, drivefile, lname
Zname, Zrate, Zssn, Zsecclr

21. ESECCONF: Prints personnel holding no clearance.
called by: ESECURE
calls: ESECURE (return)
vars: answer, driver, drivefile, lname
Zname, Zrate, Zssn, Zsecclr

22. EDELETE: Allows deletion of a enlisted form the data base.
called by: UPDATE2
calls: UPDATE2 (return)
vars: member, dropum, driver, drivefile, lname
recnum, decide, all Zenlisted

23. EFIXIT2: Permits user to change individual fields of a selected enlisted data.
called by: UPDATE2
calls: UPDATE2 (return)
vars: member, yesno, driver, drivefile, lname
recnum, decide, all Zenlisted

24. EADD: Permits user to add a new enlisted to the data base.
called by: UPDATE2
calls: UPDATE2 (return)
vars: addmore, yesno, driver, drivefile, loopit,
decide, addum

25. DATADIC1: Field description for adding to officer data base. May be viewed at user's option.
called by: OADD
calls: OADD (return)

vars: none

26. DATADIC2: Field description for adding to enlisted data base. May be viewed at user's option.

called by: EADD

calls: EADD (return)

vars: none

27. QUIT: Takes user out of Personnel Management System and back to DOS.

called by: GREET
UPDATE1
UPDATE2

calls: DOS

vars: none

2. Variables

a. ADDMORE: Used by OADD and EADD to keep adding new records.

b. ADDUM: Used by OADD and EADD to hold dummy DBF during additions.

c. ANSWER: Stores "P" if printout is desired in OSENIOR, OSECURE, ORECALL, EROSTER, ESENIOR, ERECALL, MANPOWER, ESECALL, ESECTOP, ESECSEC, ESECCONF, and ESECNONE.

d. CNT: Counts to see if anyone is found in OSENIOR and OSECURE.

e. DECIDE: Holds "M", "C", "D", or "P" while processing in OADD, ODELETE, EADD, and EDELETE.

f. DRIVEFILE: Holds drive number and file being used.

g. DRIVER: Holds the drive number of the data files.

h. DROPUM: Used by ODELETE and EDELETE to continue deleting records.

i. ENL: Used by GREET to make sure ENLISTED.DBF is present on disk.

j. LNAME: Holds last name while searching in OSENIOR, OSECURE, ODELETE, ORECALL, OFIXIT1, EROSTER, ERECALL, EFIXIT2, and EDELETE.

k. LOOPIT: Used to verify response in GREET, OADD, EADD.

l. MEMBER: Used to hold name of desired person in ODELETE, OFIXIT1, EDELETE, and EFIXIT2.
m. OFFR: Used by GREET to make sure OFFICER.DBF is present on disk.
n. RECNUM: Holds current record number in ODELETE, OFIXIT1, EDELETE, and EFIXIT2.
o. SELECT: Holds selected menu value in GREET, MAIN1, MAIN2, UPDATE1, and UPDATE2.
p. SELECTNUM: Holds numerical value of selected menu value in GREET, MAIN1, MAIN2, UPDATE1, and UPDATE2.
q. YESNO: Holds "Y" or "N" choice in OADD, OFIXIT1, EADD, and EFIXIT2.
r. ZRATE: Holds member's rate.
s. ZBPHONE: Holds member's business phone.
t. ZBSC: Holds member's billet sequence code.
u. ZCITY: Holds member's hometown.
v. ZDEPT: Holds member's assigned department.
w. ZDOR: Holds member's date of rank (YYMMDD).
x. ZEAOS: Holds end date of member's obligated service (YYMMDD).
y. ZHPHONE: Holds member's home phone number.
z. ZNAME: Holds member's name (last first mi).
aa. ZPGC: Holds member's pay grade code:
E-9 = 09, E-8 = 18, E-7 = 27,
E-6 = 36, E-5 = 45, E-4 = 54,
E-3 = 63, E-2 = 72, and E-1 = 81. Used to sort by rate.
bb. ZPNEC: Holds member's primary NEC.
cc. ZPRD: Holds member's projected rotation date.
dd. ZRANK: Holds officer's rank.
ee. ZRATE: Holds enlisted's rate.
ff. ZRBSC: Holds member's reserve BSC.
gg. ZSECCLR: Holds member's security clearance.
hh. ZSNEC: Holds member's secondary NEC.

ii. ZSSN: Holds member's social security number.

jj. ZSTATE: Holds member's home state.

kk. ZSTATUS: Holds member's status: TAR, NONTAR, or SELRES.

ll. ZSTREET: Holds member's street and house number.

mm. ZZIP: Holds member's zip code: either 6 or 9 digit.

nn. ZZONE: Holds member's reserve zone (by address areas).

3. Data Base Structures

```
. use officer
. display structure
STRUCTURE FOR FILE: A:OFFICER .DBF
NUMBER OF RECORDS:  00024
DATE OF LAST UPDATE:02/26/86
PRIMARY USE DATABASE
FLD       NAME       TYPE WIDTH   DEC
001       NAME        C     022
002       RANK        C     007
003       SSN         C     013
004       STREET      C     022
005       CITY        C     017
006       STATE       C     005
007       ZIP         C     010
008       HPHONE      C     014
009       BPHONE      C     014
010       SECCLR      C     022
011       ZONE        C     004
** TOTAL **            00151
```

```
. use enlisted
. display structure
STRUCTURE FOR FILE:  A:ENLISTED.DBF
NUMBER OF RECORDS:   00052
DATE OF LAST UPDATE:02/26/86
PRIMARY USE DATABASE
FLD        NAME       TYPE WIDTH    DEC
001      BSC            C     007
002      RBSC           C     007
003      NAME           C     022
004      ARATE          C     007
005      RATE           C     007
006      SSN            C     013
007      PNEC           C     006
008      SNEC           C     006
009      EAOS           C     006
010      PRD            C     006
011      DOR            C     006
012      STATUS         C     006
013      DEPT           C     012
014      PGC            C     004
015      STREET         C     022
016      CITY           C     017
017      STATE          C     005
018      ZIP            C     010
019      HPHONE         C     014
020      BPHONE         C     014
021      SECCLR         C     022
022      ZONE           C     004
** TOTAL **             00224
```

4. Indexes Used

OFFICERS:

on name

ENLISTED:

on name

on pgc + dor

on bsc + rbsc

APPENDIX C

DETAILED SCHEDULING MODEL

This appendix provides the detailed breakdown of the Data Flow Diagrams into a Structure Chart for Example 1 in the sizing section of Chapter IV.

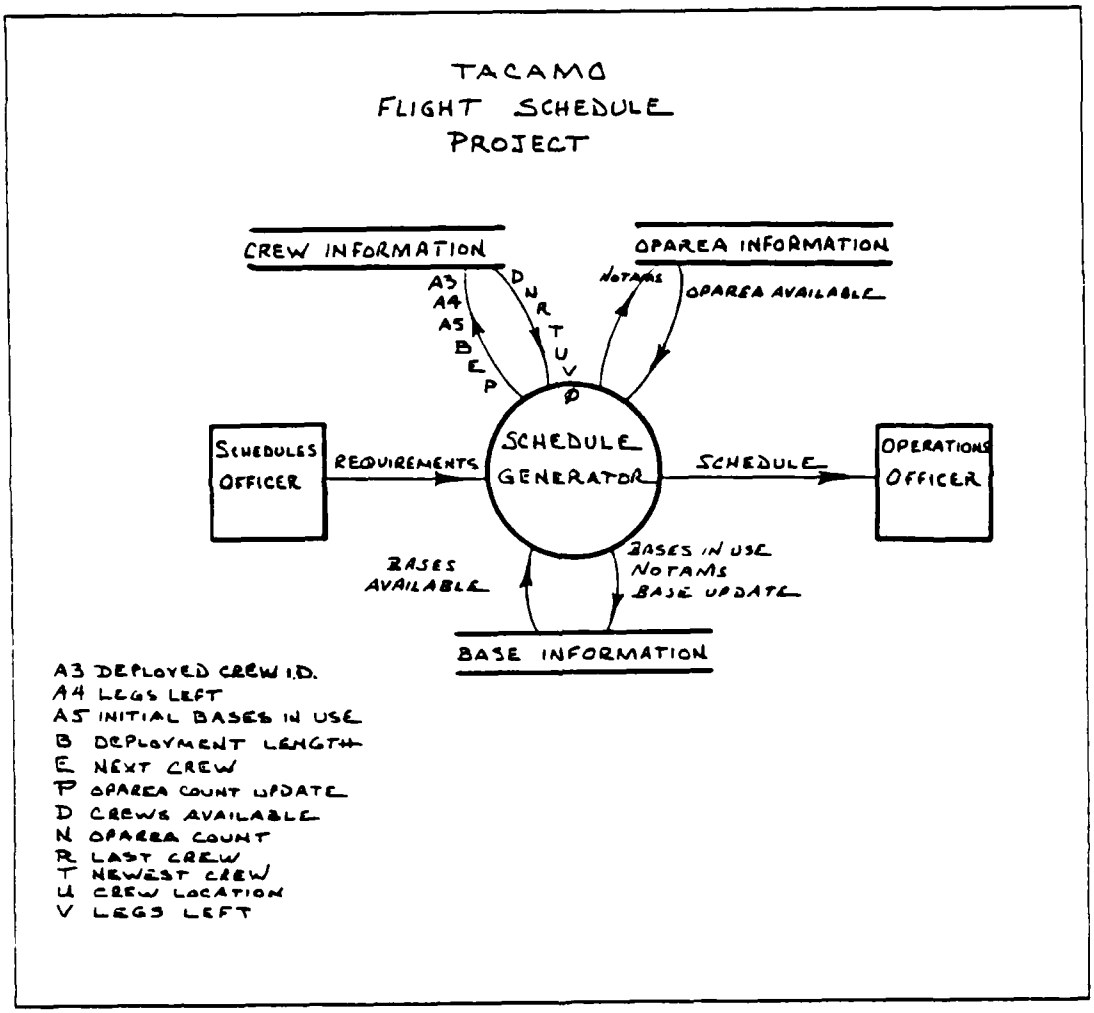


Figure C.1 Level 0 Data Flow Diagram.

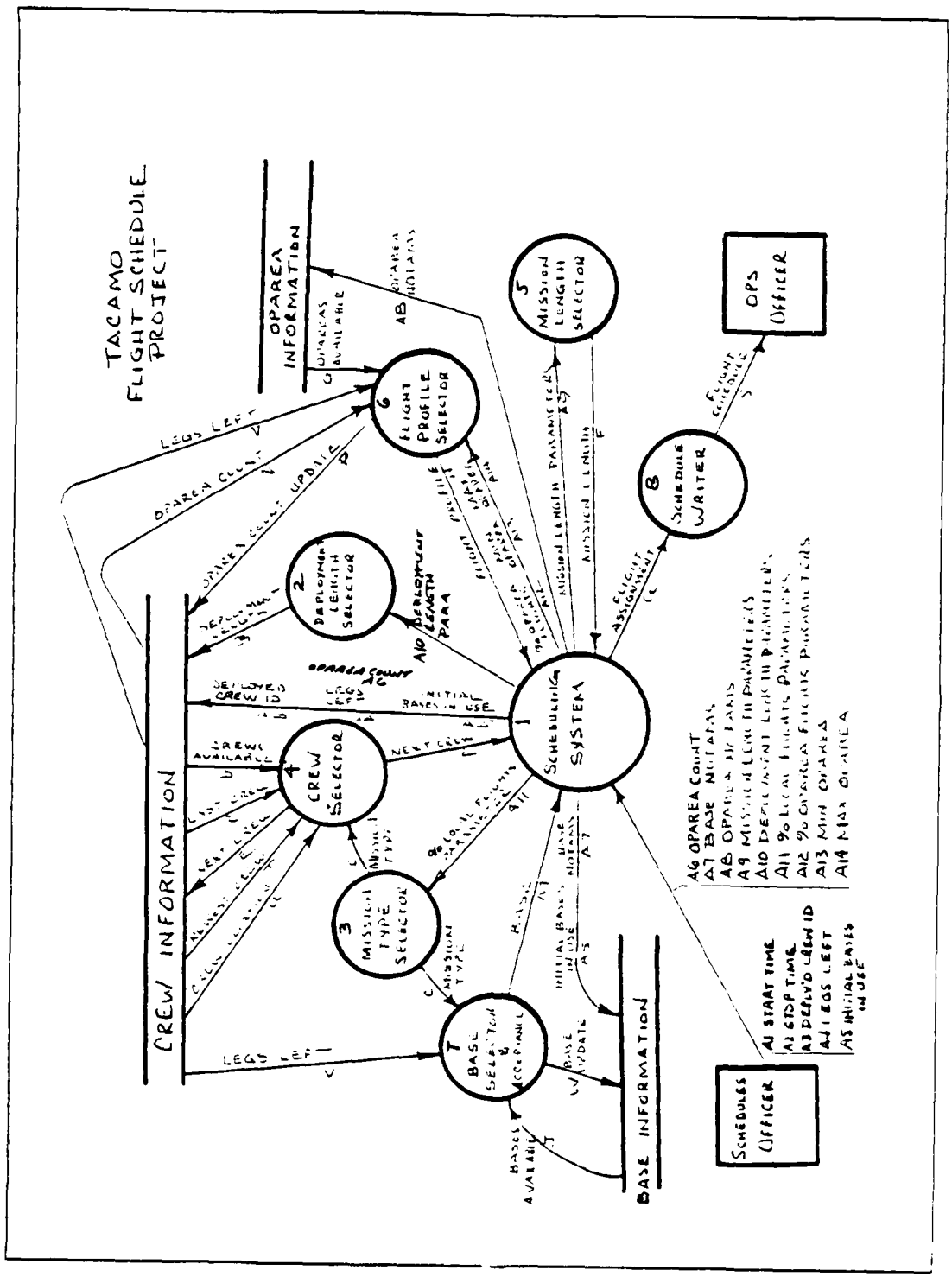


Figure C.2 Level 1 Data Flow Diagram

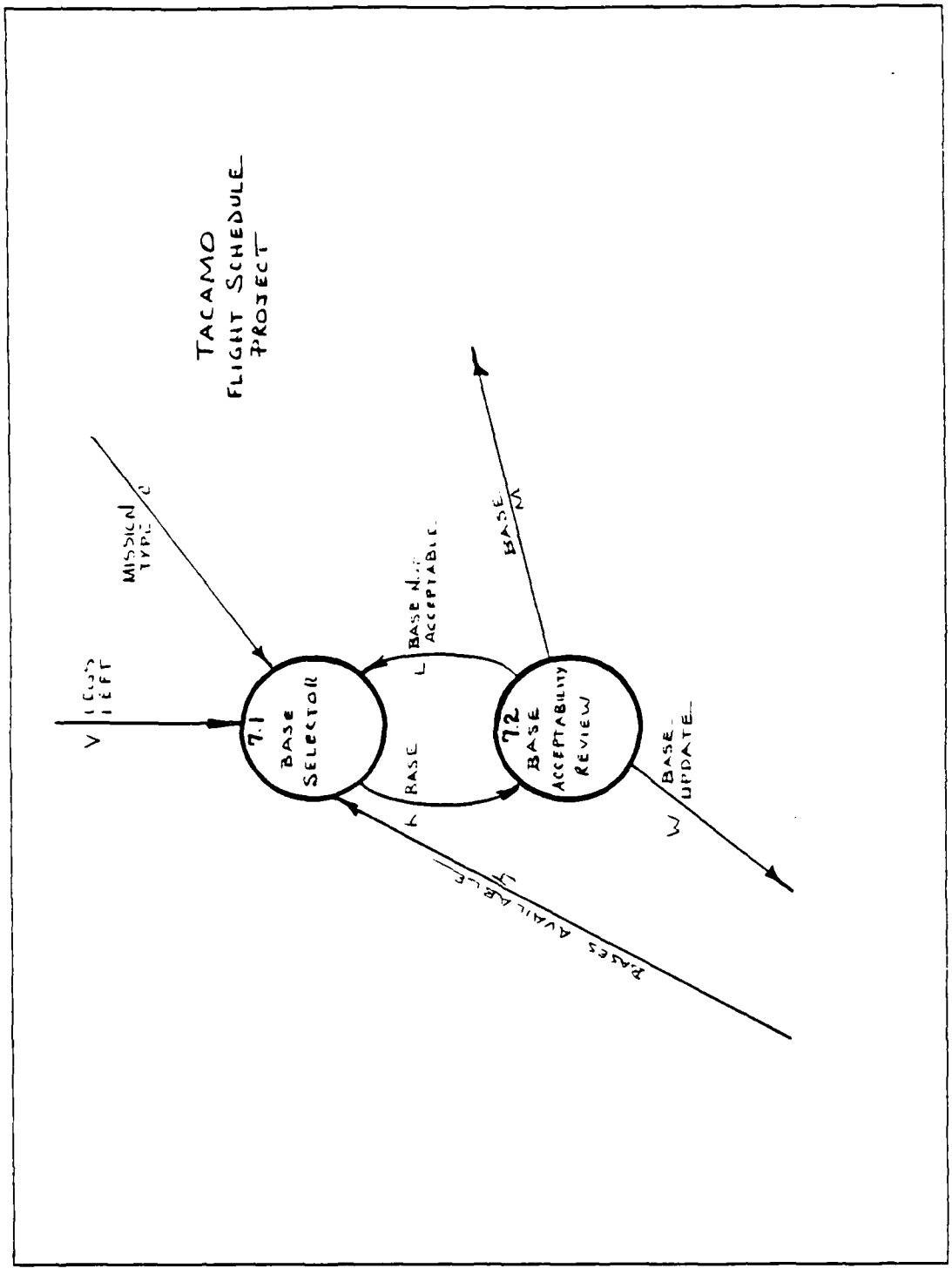


Figure C.3 Level 2 Data Flow Diagram

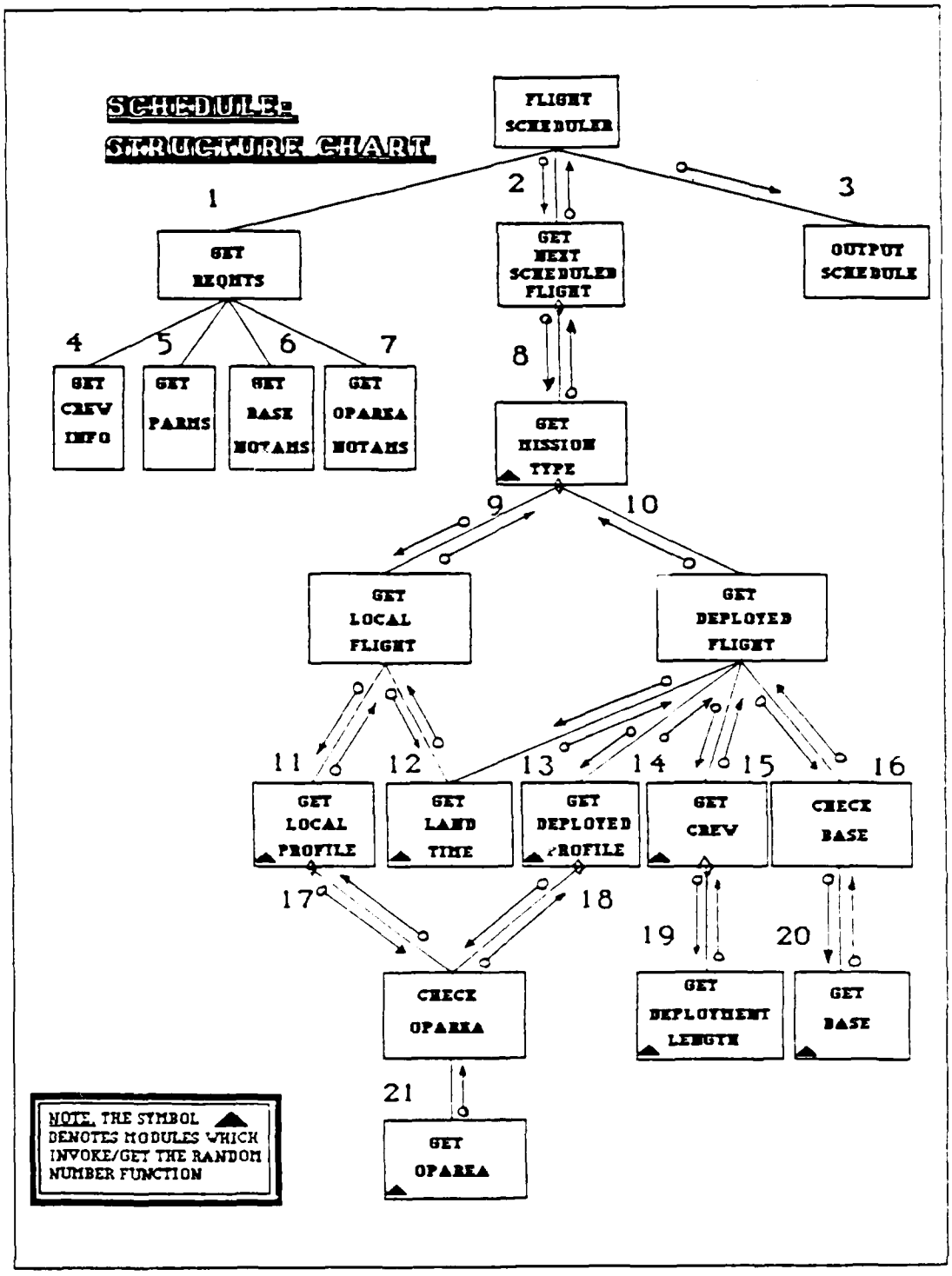


Figure C.4 Structure Chart for Scheduling Problem

LIST OF REFERENCES

1. Churchman, C. W., The Systems Approach, Dell Publishing Co., Inc., 1968.

2. Churchman, C. W. The Systems Approach and its Enemies. Basic books, Inc. 1979.

3. Wetherbe, J. C., Systems Analysis and Design, West Publishing Co., St. Paul, Minn, 1984.

4. Kramer, N. J. T. A., Systems Thinking, H. E. Stenfert Kroese B. V., Leiden, 1977.

5. Nugent, C. E. and Vollman, T. E., "A Framework for the Systems Design Process," Decision Science, 1972.

6. Wenninger, E. P., "Entropy, Equilibrium and Organizations: Problems of Conceptualization," General Systems and Organizational Theory: Methodological Aspects, Kent State University Press, 1975.

7. Cleland, D. I. Management: A Systems Approach, McGraw Hill, Inc., New York, 1972.

8. Kroenke, D. M., Business Computer Systems, Mitchell Publishing Co., Santa Cruz, CA, 1984.

9. Gibson, C. F. and Nolan, R. L., "Managing the Four Stages of EDP Growth," Harvard Business Review, JAN/FEB 1974.

10. Nolan, R. L., "Managing the Crisis in Data Processing," Harvard Business Review, MAR/APR 1979.

11. Goldstein, R. C. and McCririck, I. B., "The Stage Hypothesis and Data Administration: Some Contradictory Evidence," Proceedings of the Second Industrial Conference on Information Systems, DEC 1981.

12. McWilliams, P. A., Word Processing on the IBM, Prelude Press, Random House, 1983.

13. Froelich, R. A., The Free Software Catalog and Directory, Crown Publishing, Inc., N. Y., 1984.

14. MS-DOS Version 2, Zenith Sata Systems, Inc., 1984.

15. Jacobson, B., "What is a Data Base Package?," Creative Computing, Sept 1984.

16. Matthews, C., Word Processing, McGraw-Hill Book Co., NY, 1985.

17. Ettlin, W. A., WordStar Made Easy, Second Edition, Osborne/McGraw-Hill, 1982.

18. Lotus 123 User's Manual, Lotus Development Corp., Mass, 1983.

19. McGlynn, D. R., Personal Computing, John Wiley & Sons, NY, 1979.

20. Desktop Computer, Z-100 Series User's Manual Zenith Data Systems, Corp., 1982.

21. Senn, J. A., Analysis ans Design of Information Systems, McGraw-Hill, Inc., 1984.

22. Boehm, B. W., Software Engineering Economics, Prentice-Hall, Inc., 1981.

23. Pressman. R. S., Software Engineering: A Practitioner's Approach. McGraw-Hill, Inc., 1982.

24. DeMarco, T., Structured Analysis and System Specification, Yourdon, Inc., 1978.

25. Brod, C., Technostress, Addison-Wesley, 1984.

26. Brod, C., "Technostress," Computer World, Farmingham, Mass., 1984.

27. Hutchinson, R. A., Computer Eye-Stress, M. Evans and Co., Inc., N.Y., 1985.

28. Andrews, D.C., and Leventhal, N., "Office Automation," Government Computer News, GCN Communications Corp., Feb 1985.

29. COMNAVAIRPAC Instruction 5239.2, San Diego, CA, 5 August 1985.

30. Rochester, J. B., and Gantz, J., The Naked Computer, William Morrow and Co., Inc., N. Y., 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
2.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5000	1
3.	Computer Technology Programs Code 37 Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor M. P. Spencer, Code 54 SP Naval Postgraduate School Monterey, CA 93943-5000	2
5.	Professor J. W. LaPatra, Code 54 LP Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
7.	Lt Paul Bundy, Code 20 COMNAVAIRPAC Naval Air Station, North Island San Diego, CA 92135	1
8.	Mr. Owen Jones, Code N9 COMASWWINGPAC Naval Air Station, North Island San Diego, CA 92135	1
9.	CDR M. W. Skahan 1813 Gotham Street Chula Vista, CA 92010	3

END

DTIC

7-86